

APERFEIÇOANDO-SE NO
TK 95
COMPATÍVEL COM TK90X

Você leu *TK90X — Primeiros Passos*, e já começou a escrever seus próprios programas. Este livro o leva mais além, apresentando os mais avançados recursos do seu micro.

Descubra como desenhar (UDG) seus próprios aviões, carros e monstros espaciais e como usá-los em jogos de ação, aproveitando os espetaculares recursos de animação do TK95.

Aprenda a programar jogos excitantes e *adventures*, e divirta-se muito com os diversos programas completos apresentados no livro, que poderão ser adaptados e incrementados por você. Entre eles:

- **PARCEIROS:** Seu TK95 arranjou uma profissão — agente matrimonial.
- **CIRCUITO:** Mantenha seu carro na estrada e evite acidentes... se puder!
- **CODEMASTER:** Use seu TK95 para enviar mensagens secretas em código para seus amigos.
- **CAMPO MINADO:** Ande com cuidado e tente chegar vivo ao castelo.

CAROLYN HUGHES

professora inglesa, tanto em cursos primários quanto secundários, e uma especialista no uso do ZX81 e do ZX Spectrum (compatíveis com o TK85, o TK90X e o TK95) na Educação.

ISBN 85-7001-385-X

(Edição original: ISBN 0-00-692366-6 A Fontana Original, London, England.)

HUGHES

CAROLYN HUGHES

APERFEIÇOANDO-SE NO TK95

APERFEIÇOANDO-SE NO

TK 95

COMPATÍVEL
COM TK90X

Editora Campus

APERFEIÇOANDO-SE NO

TK 95

COMPATÍVEL COM TK90X

Litec

LIVRARIA EDITORA TÉCNICA LTDA

Rua dos Timbiras 257 - CEP: 01208 - São Paulo
Caixa Postal 30 869 - Tel.: 222-0477

REF. 3173 PREÇO 4.300,00

CAROLYN HUGHES

APERFEIÇOANDO-SE NO

TK 95

COMPATÍVEL COM TK90X

Ilustrações

Bryan Reading

Tradução

Astrid Beatriz de Figueiredo

Tradutora/Intérprete — Faculdade Estácio de Sá

Este livro foi impresso nas oficinas gráficas da
Editora Vozes Ltda.,
Rua Frei Luís, 100 — Petrópolis, RJ,
com filmes e papel fornecidos pelo editor.

Editora Campus Ltda.

Rio de Janeiro

Do original:
Second Steps With your SPECTRUM
Copyright © 1984, by Fontana Paperbacks.
All Rights Reserved.

© 1987, Editora Campus Ltda.

Todos os direitos para a língua portuguesa reservados e protegidos pela
Lei 5988 de 14/12/1973.

Nenhuma parte deste livro poderá ser reproduzida ou transmitida sejam
quais forem os meios empregados: eletrônicos, mecânicos, fotográficos,
gravação ou quaisquer outros.

Todo o esforço foi feito para fornecer a mais completa e adequada
informação. Contudo a editora e o(s) autor(es) não assumem
responsabilidade alguma pelos resultados e uso da informação fornecida.
Recomendamos aos leitores, em consequência, testar toda a informação
antes de sua efetiva utilização.

Capa
Otavio Studart

Projeto Gráfico, Composição e Revisão
Editora Campus Ltda.

Qualidade internacional a serviço do autor e do leitor nacional.

Rua Barão de Itapagipe 55 Rio Comprido
Tel.: (021) 284 8443 Telex (00038) 021-32606
20261 Rio de Janeiro RJ Brasil
Endereço Telegráfico: CAMPUSRIO

ISBN 85-7001-385-X

(Edição original: ISBN 0-00-692366-6 A Fontana Original, London.)

Ficha Catalográfica
CIP-Brasil. Catalogação-na-fonte.
Sindicato Nacional dos Editores de Livros, RJ.

H888a Hughes, Carolyn
Aperfeiçoando-se no TK95 / Carolyn Hughes; tra-
dução Astrid Beatriz de Figueiredo. — Rio de Janeiro:
Campus, 1987.

Tradução de: Second steps with your spectrum.
ISBN 85-7001-385-X

1. Programação (Computadores eletrônicos) 2. TK95
(Computadores eletrônicos). I. Título.

86-0864

CDD — 001.64
CDU — 681.3

Aperfeiçoando-se no TK95

O TK95 é um equipamento versátil e uma vez que você
tenha aprendido a fazer programas simples, certamente
gostará de continuar progredindo, e usar cada vez mais as
características especiais do seu computador. Este livro
desvendará os mistérios das sub-rotinas e das matrizes,
mostrará como desenhar seus próprios caracteres, a usar
gráficos de alta resolução — e muito mais.

Em todos os capítulos foram incluídos programas para que
você possa tirar o máximo proveito das capacidades do seu
TK95 e em todo o livro você encontrará idéias e sugestões
para programas que você pode fazer usando seus novos
conhecimentos sobre programação.

Este é um livro para leitores de qualquer idade, que saibam
um pouco sobre programação em BASIC e que desejam
aprender um pouco mais de modo prático e um pouco
menos técnico.

Agradecimentos

Gostaria de agradecer a Pam, Karen e Emma pela leitura do manuscrito e pelos comentários valiosos, e a Andrew pelas sugestões e assistência em relação aos programas maiores.

Sobre a Autora

Carolyn Hughes travou conhecimento com o mundo dos computadores quando sua família adquiriu um Sinclair ZX80. Desde então foram adquiridos um ZX81, um ZX Spectrum e um Micro BBC.

A autora tem três filhos e preside um clube de computadores na escola onde estuda seu filho mais novo. Embora esteja especialmente interessada no envolvimento de crianças com computadores, tanto para o aprendizado quanto como divertimento, ela compreende que os adultos também precisam de manuais que não sejam extremamente técnicos para auxiliá-los em seus primeiros contatos com os computadores, e mais tarde para desenvolver sua capacidade de programar.

Ela é a autora de *First Steps with Your Spectrum* publicado pela Editora Fontana e *First Steps with Your BBC Micro* e produz também software educacional

Notas da Edição Brasileira:

Os micros ingleses ZX80, ZX81 e ZX Spectrum são compatíveis, respectivamente, com os nacionais TK80, TK85 e TK95/TK90X.

O livro *First Steps With Your Spectrum* foi publicado pela Editora Campus sob o título *TK90X — Primeiros Passos*.

Índice

- 1. APERFEIÇOANDO-SE NO TK95, 11**
Alguns lembretes. "Areias Movediças"
- 2. O TK95 INTERPRETA OS DADOS, 15**
READ and DATA. "Early one Morning"
- 3. MAIS DADOS, 18**
RESTORE. Um pouco mais de música e um cartão de aniversário musical
- 4. BLOCOS MAIORES, 23**
DIM para matrizes de números
- 5. COMPARTIMENTOS PARA CARACTERES, 27**
Matrizes do tipo string
- 6. UM POUCO MAIS SOBRE MATRIZES DO TIPO STRING, 30**
Blocos maiores com várias linhas de compartimentos
- 7. O TK95 MANTÉM UM REGISTRO, 33**
Registros e campos. Números de telefone — um programa
- 8. PARA LÁ E PARA CÁ, 38**
Sub-rotinas. GO SUB e RETURN. "Perguntas e Respostas"
- 9. MARQUE UM PONTO E TRACE UMA LINHA, 43**
Gráficos de alta resolução. Figuras e formas com PLOT e DRAW
- 10. CÍRCULOS, 49**
Desenhos com CIRCLE
- 11. O TK95 INDICA UM PONTO, 53**
Procurando um ponto com POINT. "Garatujas"
- 12. A SUA IMPRESSORA ZX, 58**
COPY PRINT e LLIST com a impressora
- 13. CARACTERES E CÓDIGOS, 64**
CODE e CHR\$. Uma olhada nas palavras e caracteres do TK95
- 14. SEUS PRÓPRIOS CARACTERES, 67**
Como projetar e programar seus próprios caracteres gráficos
- 15. ESBARREI EM ALGUMA COISA?, 74**
Descobrimo colisões com SCREEN\$. "Paraquedista"
- 16. EFEITOS ESPECIAIS, 80**
Realce seus programas com FLASH, BRIGHT, INVERSE e OVER
- 17. ATRIBUTOS, 87**
Como descobrir uma colisão com um caractere gráfico. ATTR. "Elefante"
- 18. MAIS TAREFAS PARA SEU GRAVADOR, 92**
Como salvar um programa para que ele entre automaticamente ao ser carregado. Salvando uma tela. "Arte Moderna"
- 19. MEDIR E CORTAR UM STRING, 97**
Medindo uma palavra com LEN
- 20. BUGS, 102**
Como detectar erros que interferem com seus programas. Mensagens de erros

21. CODEMASTER, 106

Um programa para transformar o seu TK95 em uma máquina de codificar e de decifrar códigos

22. CAMPO MINADO, 111

Atravesse o pátio do castelo — se puder!

23. PARCEIROS, 116

Talvez o seu TK95 descubra o parceiro ideal

24. CIRCUITO, 122

Um programa que testa sua habilidade como motorista

25. AVENTURA NA SELVA, 127

Tente esta e depois escreva outra aventura

26. CURSORES E TECLAS — UM LEMBRETE, 134

Dicas para usar o teclado do seu TK95

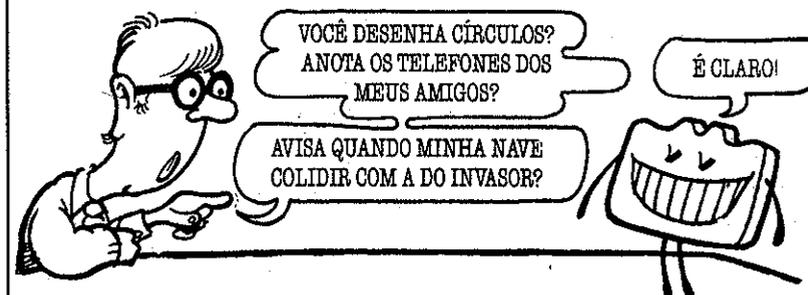
27. SUGESTÕES, 138

Idéias para resolver os quebra-cabeças da seção. Para você fazer

1

Aperfeiçoando-se no TK95

Este livro é destinado às pessoas que sabem como fazer programas simples, e que desejam ir um pouco mais além — fazer programas mais complicados e utilizar melhor as características especiais do TK95



Como você sabe, a linguagem do TK95 é o BASIC. Se você examinar bem o teclado, verá em branco, verde e vermelho, todas as palavras em BASIC que ele entende, e que você pode utilizar para controlar o computador. Provavelmente você já sabe como utilizar cerca de vinte ou trinta dessas palavras.

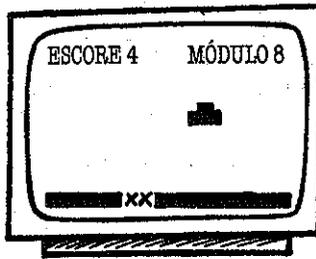
Aqueles que aprenderam a programar em outros computadores precisarão conhecer a família de cursores do TK95, o que lhes permitirá digitar uma palavra em BASIC simplesmente pressionando uma tecla.



Você pode aprender mais a respeito deles no Capítulo 26.

Você verá quatro destes cursores em ação quando digitar "Areias Movediças". O programa utiliza a cor e o som do TK95, e inclui cerca de outras quinze palavras de instrução em BASIC.

Você é um piloto espacial. Sua missão é fazer pousar dez pequenos módulos de aterrissagem em um planeta estranho. Infelizmente a única pista de pouso é um terreno de areias movediças. Para conduzir seu módulo para a esquerda, pressione a tecla 5, e para conduzi-lo para a direita pressione a tecla 8.



Seu módulo é representado por dois caracteres gráficos ■ e ■. Com o cursor G na tela, mantenha pressionada CAPS SHIFT e digite 2 e depois 1. Você também precisa do cursor G para a areia (linhas 90, 100 e 180). Desta vez mantenha pressionada CAPS SHIFT e digite B.

```
10 REM areias movedicas
20 LET c$=" ■ "
30 BORDER 1: PAPER 1: CLS
40 LET score=0
50 FOR t=1 TO 10
60 LET yc=16
70 LET pc=INT (RND*31)
```

```
80 INK 6
90 PRINT AT 19,0;" ■■■■■ "
■■■■■ "
100 PRINT AT 20,0;" ■■■■■ "
■■■■■ "
110 PRINT AT 0,24;"modulo";t
120 INK 4
130 PRINT AT 19,pc;" ■";AT 20,p
c;"XX"
140 INK 7
150 PRINT AT 1,yc;c$
160 FOR r=2 TO 19
170 INK 6
180 PRINT AT 19,pc;" ■";AT 20,p
c;" ■"
190 LET a=INT (RND*4)+1
200 IF a=1 AND pc>0 THEN LET p
c=pc-1
210 IF a=2 AND pc>30 THEN LET
pc=pc+1
220 INK 4
230 PRINT AT 19,pc;" ■";AT 20,p
c;"XX"
240 LET d$=INKEY$
250 PRINT AT r-1,yc;" ■ "
260 IF d$="5" AND yc>0 THEN LET yc=yc-1
270 IF d$="8" AND yc<30 THEN LET yc=yc+1
280 INK 7
290 PRINT AT r,yc;c$
300 SOUND .1,0
310 NEXT r
320 IF pc=yc THEN FOR b=0 TO 1
2: SOUND .1,b: NEXT b: LET score=
score+1
```

```

330 IF pc < >yc THEN FOR b=0 TO
-12 STEP -1: SOUND .1,b: NEXT b
340 PRINT AT 0,0;"escore ";escore
350 NEXT t
360 BORDER 7: PAPER 7: INK 0

```

Como você possui dez módulos de aterrissagem, a maior parte do programa é repetida dez vezes. O TK95 emprega um loop FOR...NEXT, com uma variável denominada t para contar de 1 a 10. O loop se inicia na linha 50:

```

50 FOR t = 1 TO 10
e termina na linha 350:
350 NEXT t

```

Neste loop existe um outro para contar as linhas na tela à medida que o módulo de aterrissagem desce, pouco a pouco, em direção à pista. O início deste loop é

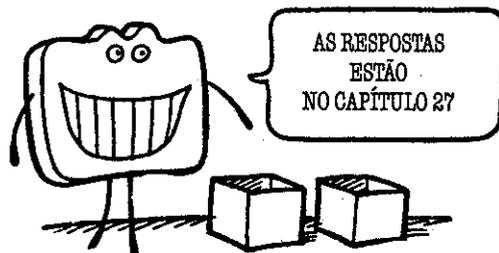
```

160 FOR r = 2 TO 19
e o final
310 NEXT r

```

Para você fazer

Faça uma lista das variáveis numéricas e das variáveis de string que existem no programa. Cada uma delas aparece primeiro em uma linha começando com a palavra LET. Veja se você é capaz de descobrir para que serve cada uma delas.



2

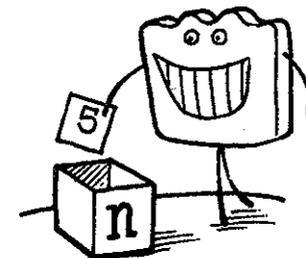
O TK95 Interpreta os Dados

Até agora você travou conhecimento com duas maneiras de colocar um número em um bloco de variáveis numéricas:

```
INPUT n
```

e

```
LET n = 5
```



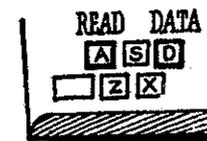
Existe uma outra maneira, utilizando as palavras READ e DATA. Você dá ao TK 95 uma lista de informações (denominada lista de dados). O computador lê os itens da lista, e coloca cada um em um bloco de variável.

Aqui está um exemplo:

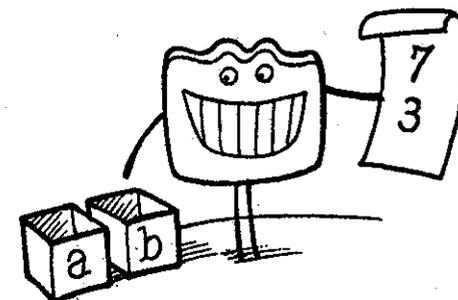
```

10 READ a,b
20 PRINT "No bloco a tenho ";a
30 PRINT "No bloco b tenho ";b
40 DATA 7,3

```



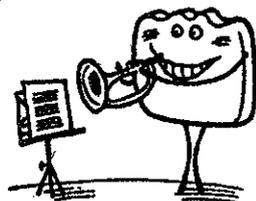
S.S.S.—2



Você pode colocar a lista de dados em qualquer parte do programa que desejar. Entretanto é uma boa idéia colocá-la no final, para que seja fácil encontrá-la quando você quiser alterar os dados.

Aqui está um programa em que o TK95 utiliza a informação em uma lista de dados para tocar uma melodia.

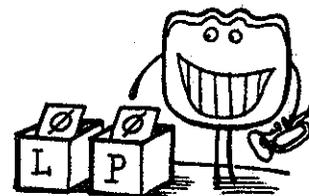
```
10 PRINT "EARLY ONE MORNING"  
20 READ L,P  
30 IF L=0 THEN STOP  
40 SOUND L,P  
50 GO TO 20  
100 DATA 1,5,.5,5,.5,5  
110 DATA .5,5,.5,9,.5,12,.5,12  
120 DATA .5,14,.5,10,.5,7,.5,7  
130 DATA .5,4,.5,7,.5,0,.5,0  
140 DATA 1,5,.5,5,.5,5  
150 DATA .5,5,.5,9,.5,12,.5,12  
160 DATA .5,14,.5,10,.5,7,.5,4  
170 DATA 2,7  
180 DATA 0,0
```



CADA NÚMERO NA LISTA DE DADOS DEVE SER SEGUIDO DE UMA VÍRGULA EXCETO O ÚLTIMO, NO FINAL DA LINHA

Na linha 20, o TK95 lê os dois primeiros números da lista de dados e os insere nos blocos L e P. Na linha 40, ele utiliza estes dois números para tocar uma nota musical. L é a duração da nota e P é o seu diapasão (a altura do tom).

Em seguida ele volta e lê os próximos dois números. Você vai perceber que os dois últimos números na lista de dados são 0,0. Quando o computador vê esses números, sabe que a melodia acabou. (Veja linha 30.)



Você poderia colocar todos os dados em uma longa linha de programa porém a digitação é mais fácil se os dados estiverem em seções. No programa "Early one Morning", cada linha de dados é um compasso da música.

3

Mais Dados

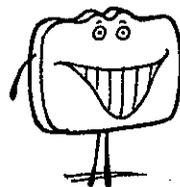
Você pode também colocar palavras em blocos de variável de string com READ e DATA.

Aqui está um programa que servirá de exemplo:

```
10 READ a$,b$
20 PRINT "Acabo de ler"
30 PRINT a$ "e" b$
40 DATA "chocolate", "creme"
```

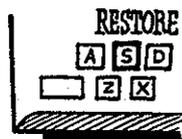


DADOS EM SÉRIE DEVEM VIR ENTRE ASPAS



VOCÊ SABE PARA QUE SERVEM OS APÓSTROFOS DA LINHA 30?

RESTORE é uma palavra que acompanha READ e DATA.



Usa-se RESTORE para dizer ao computador onde ele deve começar a ler na lista de dados. A palavra RESTORE sozinha significa "começar a ler no início da lista de dados". Se você colocar um número depois de RESTORE, o computador começará a ler os dados a partir da linha que tenha aquele número. Por exemplo, RESTORE 200 quer dizer "comece a ler os dados a partir da linha 200".

Neste programa o computador pode tocar duas melodias. Ele vai perguntar qual das duas você quer escutar e vai então utilizar os dados para tocar aquela melodia.

```
10 PRINT "Eu posso tocar"
20 PRINT AT 8,0;"1 Loch Lomon"
d"
30 PRINT AT 10,0;"2 Lullaby"
40 INPUT "Digite o número de sua
escolha.":c
50 CLS
```

As linhas de 10 a 50 formam a primeira tela completa do programa, onde o computador pergunta qual das melodias você quer escutar.



```
60 IF c=1 THEN RESTORE
70 IF c=2 THEN RESTORE 300
80 IF c < > 1 AND c < > 2 THEN RUN
```

Estas três linhas verificam o que está contido em c. Se não for 1 ou 2, o computador rodará outra vez todo o programa.



```

90 READ a$
100 PRINT "Estou tocando" 'a$
110 READ n
120 FOR a=1 TO n
130 READ d,p
140 SOUND d,p
150 NEXT a
    
```



ESTA É A PARTE QUE TOCA A MELODIA
a\$ É O NOME DA MELODIA E n É O NÚMERO DE NOTAS A SER TOCADO

```

200 DATA "Loch Lomond",20
210 DATA 1,2
220 DATA 1,7,.75,7,.25,9,1,11,.
5,9,.5,7
230 DATA 1,9,.75,9,.25,7,1,4,.5
,2;.5,2
240 DATA 1,7,1,7,1,7,.5,11,.5,1
4
250 DATA 2,16,1,14
    
```

```

300 DATA "Lullaby",27
310 DATA .5,9,.5,9
320 DATA 1.5,12,.5,9,1,9
330 DATA 2,12,.5,9,.5,12
340 DATA 1,17,1.5,16,.5,14
350 DATA 1,14,1,12,.5,7;.5,9
360 DATA 1,10,1,7,.5,7;.5,9
370 DATA 2,10,.5,7,.5,10
380 DATA .5,16,.5,14,1,12,1,16
390 DATA 1,17
    
```



Para você fazer

Talvez você queira fazer um "cartão de aniversário musical" para alguém. Aqui está a melodia "Happy Birthday to you". O número que está abaixo de cada nota representa os valores da tonalidade no TK95.

tem comprimento 2.

 tem comprimento 1 e

 é meio compasso (.5).

Hap py Birth day to you
 Hap py Birth day to you
 Hap py Birth day dear Em ma
 Hap Py Birth day to you

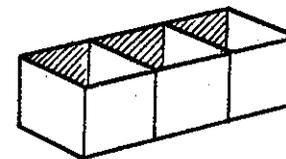
A tela será o cartão com as palavras impressas, e você pode enfeitar as bordas com estrelas coloridas (ou até mesmo se quiser, alguma coisa mais sofisticada). Veja se você é capaz de fazer o computador imprimir as palavras linha por linha, tocando a melodia correspondente a cada linha antes que a seguinte apareça.

Existem várias maneiras interessantes de escrever este programa. Você encontrará algumas sugestões no Capítulo 27.

4

Blocos Maiores

Às vezes a capacidade que um programa tem de armazenar vários números juntos, num grande bloco com várias subdivisões, pode ser muito útil.



Para criar um destes blocos na memória do TK95, você deve dizer a ele o nome do bloco e definir seu tamanho. A palavra em BASIC usada para isto é DIM.



Se você digitar

DIM a(4)

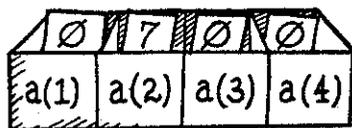
o TK95 ficará sabendo que você deseja um bloco denominado a, com 4 compartimentos. Este tipo de bloco é

denominado "array" (matriz). Seu nome deve ser uma letra do alfabeto.



Inicialmente todos os compartimentos são indicados pelo 0. Você pode alterar os números utilizando LET, INPUT ou READ. Suponha que você deseja o número 7 no segundo compartimento. Você poderia digitar:

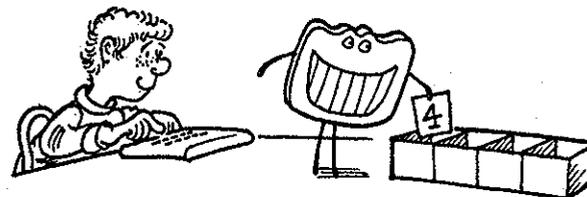
```
LET a(2)=7
```



Aqui está um programa para mostrar uma matriz sendo preenchida utilizando INPUT:

```
10 DIM b(5)
20 PRINT "Digite 5 numeros"
30 PRINT "Pressione ENTER apos cada
um deles"
40 PRINT
50 FOR n=1 TO 5
60 INPUT b(n)
70 PRINT "Coloquei";b(n)
```

```
80 PRINT "em um compartimento"
;n
90 PRINT
100 NEXT n
```



No próximo programa, o TK95 lê os números em uma matriz em uma lista de dados. Depois disto, ele pede que você escolha um número e vai então procurar na matriz para verificar se o número que você escolheu está em algum dos compartimentos.

```
10 DIM a(6)
20 FOR b=1 TO 6
30 READ a(b)
40 NEXT b
50 DATA 59,41,5,99,3,6

60 PRINT "De-me um numero"
70 INPUT n
80 PRINT

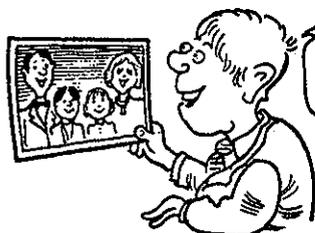
90 FOR c=1 TO 6
100 IF a(c)=n THEN PRINT "ENCONTREI!"; STOP
110 NEXT c
120 PRINT "Nao encontrei";n
```



USAMOS UM LOOP FOR...NEXT
PARA CONSEGUIR QUE O TK95
PROCURE EM CADA UM DOS
COMPARTIMENTOS

Para você fazer

Diga ao computador para criar uma matriz denominada 'a'
e nela arquivar as idades dos membros de sua família.



VOU PRECISAR DE
QUATRO COMPARTIMENTOS

O TK95 vai então perguntar a sua idade. Você a digita e ela será arquivada em um bloco denominado **eu**. Ele vai então procurar na matriz e contar quantas pessoas são mais velhas que você. Você vai precisar de um bloco de variáveis numéricas para ser utilizado pelo computador na contagem de quantos são mais velhos que você. É importante verificar se ele contém o 0 antes que o TK95 comece a pesquisar na matriz. Cada vez que ele encontrar uma idade maior que a sua, acrescenta uma ao número no bloco.

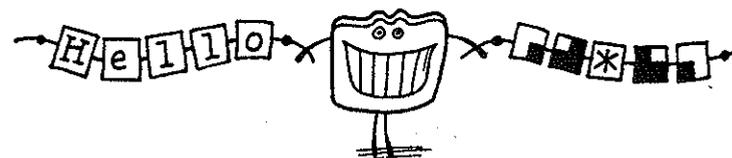
ESTOU USANDO ESTE
COMPARTIMENTO
PARA MANTER UM REGISTRO DE
QUANTAS
IDADES SÃO > DO QUE A MINHA



5

Compartimentos para Caracteres

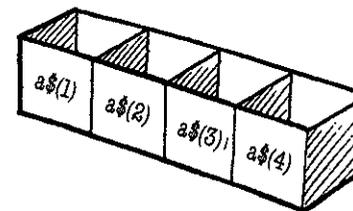
Você deve lembrar que uma string é composta de caracteres. Por exemplo, "Hello" é uma palavra composta dos caracteres, H, e, l, l e o.



Você pode criar um bloco na memória do TK95 onde cada caractere seja colocado em um compartimento. Digite:

DIM a\$(4)

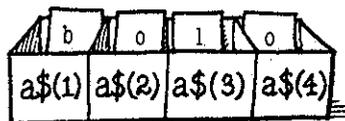
Você tem agora uma matriz denominada a\$ com quatro compartimentos.



A isto dá-se o nome de matriz do tipo string e inicialmente existe um espaço em cada compartimento. Para substituir estes espaços por letras digite:

LET a\$= "bolo"

e o computador coloca uma letra da palavra em cada um dos compartimentos.



Você pode verificar digitando:

```
PRINT a$(3)
```

O TK95 deverá mostrar a terceira letra da palavra.

O que você acha que aconteceria, se você tentasse colocar uma palavra de cinco letras em uma matriz do tipo string que tivesse somente 4 compartimentos? Diga ao computador para colocar "porta" em a\$

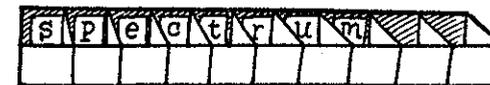
```
LET a$="porta"
```

Peça agora que ele apresente a\$ e veja o que acontece.



Vamos estabelecer agora uma matriz do tipo string com dez compartimentos e colocar nela uma palavra de oito letras. Desta vez isto será feito através de um programa:

```
10 DIM b$(10)
20 LET b$="spectrum"
```



O que há nos compartimentos nove e dez? Digite o resto do programa e faça-o rodar

```
30 FOR s=1 TO 20
40 PRINT b$;
50 NEXT s
```

Você vai ver que b\$ não é realmente "spectrum", na verdade será "spectrum ".



QUANDO VOCÊ FOR ESCOLHER UM NOME PARA UMA MATRIZ DO TIPO STRING LEMBRE-SE O NOME DEVE SER UMA LETRA DO ALFABETO SEGUIDA PELO SÍMBOLO \$

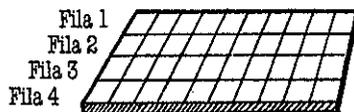
Para você fazer

Diga ao TK95 para estabelecer uma matriz do tipo string com vinte compartimentos. Você talvez queira denominá-la w\$. O computador pede para colocar uma palavra, que vai em w\$. Faça com que o computador verifique cada compartimento, contando quantas vezes a letra 's' foi encontrada.

6

Um Pouco Mais Sobre Matrizes do Tipo String

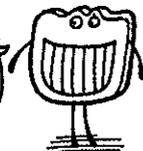
Aqui está uma matriz do tipo string bem grande. Seu nome é w\$. Ela tem espaço para quatro palavras sendo que cada uma destas palavras pode ter até dez letras.



Para dizer ao TK95 que estabeleça este bloco digite:

```
DIM w$(4,10)
```

(4,10) ME DIZ QUE EXISTEM QUATRO PALAVRAS, CADA UMA PODE TER ATÉ 10 LETRAS



Usaremos READ para colocar as palavras na matriz. A primeira palavra irá para a linha w\$(1), a segunda para a linha w\$(2), e assim por diante. Neste programa o TK95 lê os nomes de quatro árvores nas quatro linhas do bloco:

```
10 DIM w$(4,10)  
20 FOR r=1 TO 4  
30 READ w$(r)  
40 PRINT w$(r); "esta na linha"
```

```
;r
```

```
50 NEXT r  
60 DATA "pinheiro","ipe","carval  
ho","jacaranda"
```

Quando você rodar o programa verá na tela:

```
pinheiro está na linha 1  
ipê está na linha 2  
carvalho está na linha 3  
jacarandá está na linha 4
```



POR QUE EXISTEM ESPAÇOS ENTRE OS NOMES DAS ÁRVORES E AS OUTRAS PALAVRAS?



CADA LINHA DA MINHA MATRIZ TEM DEZ COMPARTIMENTOS. AQUELES QUE NÃO FORAM PREENCHIDOS POR LETRAS FICAM COM ESPAÇOS. EU MOSTREI OS ESPAÇOS

Você poderia ter colocado as palavras na matriz usando INPUT em vez de READ. Aqui está o programa outra vez mas com as alterações necessárias:

```
10 DIM w$(4,10)  
20 FOR r=1 TO 4  
30 INPUT "De-me o nome de uma ar  
vore";w$(r)  
40 PRINT w$(r);"esta na fila";r  
50 NEXT r
```

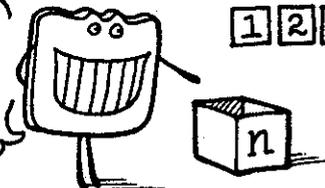
Para você fazer

Faça um programa no qual o computador estabeleça uma matriz do tipo string, denominada a\$, com espaço suficiente para cinco palavras de dez letras. Ele vai pedir cinco vezes que você digite o nome de um animal. Cada um deles vai para uma linha da matriz.

a\$(1)	e	e	f	a	n	t	e		
a\$(2)	b	o	i						
a\$(3)	r	a	t	o					
a\$(4)	p	a	n	t	e	r	a		
a\$(5)	m	i	c	o	-	l	e	ã	o

Agora o TK95 vai escolher qual dos nomes ele gosta mais. Você vai precisar usar $\text{INT}(\text{RND} * 5) + 1$ para fazer com que ele escolha um número entre 1 e 5.

VOU COLOCAR UM NÚMERO
AQUI, ELE É O NÚMERO
DO ANIMAL QUE
GOSTO MAIS



7

O TK95 Mantém um Registro

Aqui está a página de uma agenda onde alguém anotou o nome dos amigos e os números de telefone.

	Nome						Número			
1	ANNE	SMITH					709	723		
2	PETER	MORGAN					57-	0902		
3	JANET	MARTIN					304	982		
4	GORDON	MACLEAN					993	047		
5	JOHN	ANDREWS					51-	1234		
6	MATHEW	DEE					431	990		
7	SARAH	JONES					222	334		
8	WILLIAM	SMALLER					787	847		
9	GARY	MORRIS					23-	7142		
10	STEPHEN	ROGERS					101	094		

A informação foi anotada num formato semelhante ao de uma tabela, com dez linhas, uma para cada pessoa. Cada uma destas linhas é considerada um "registro".



O REGISTRO NÚMERO UM
DÁ INFORMAÇÃO A RESPEITO DE
ANNE SMITH

Cada registro tem duas partes — o nome e o número do telefone. Demos a estas partes o nome de "campos".

OS NOMES ESTÃO NO CAMPO 1
E OS NÚMEROS DE TELEFONE
NO CAMPO 2



Podemos armazenar esta informação na memória do TK95 numa matriz do tipo string. Depois que escolhermos um nome é preciso que três coisas sejam ditas ao computador para que ele possa criar um bloco do tamanho certo. Ele precisa saber quantos registros desejamos armazenar, quantos campos cada um deve ter e o tamanho do maior dos strings que vai ser armazenado:

```
DIM A$(10,2,15)
```

criará uma matriz com um tamanho adequado a essa lista de números de telefone.



O MAIS LONGO DOS STRINGS É
"WILLIAM SMALLER" QUE TEM
QUINZE CARACTERES
INCLUINDO
O ESPAÇO

Aqui está o programa que coloca os nomes e os números de telefone na matriz. Todas as letras são maiúsculas portanto coloque seu TK95 na modalidade CAPS LOCK pressionando CAPS SHIFT e CAPS LOCK. Ao executar o programa continue na modalidade CAPS LOCK.

```
10 DIM A$(10,2,15)
20 FOR R = 1 TO 10
30 READ A$(R,1),A$(R,2)
40 NEXT R
50 DATA "ANNE SMITH","709723"
60 DATA "PETER MORGAN","51-090
2"
70 DATA "JANET MARTIN","304982
"
```

```
80 DATA "GORDON MACLEAN","9930
41"
90 DATA "JOHN ANDREWS","51-123
4"
100 DATA "MATTHEW DEE","431990"
110 DATA "SARAH JONES","222334"
120 DATA "WILLIAM SMALLER","787
841"
130 DATA "GARY MORRIS","23-1142"
140 DATA "STEPHEN ROGERS","1010
94"
```

Vamos agora ampliar o programa para que o computador possa encontrar determinado número quando precisarmos dele. Em primeiro lugar o TK95 vai pedir que você digite um nome:

```
150 PRINT "QUAL O NUMERO QUE VOCE
PRECISA?"
```

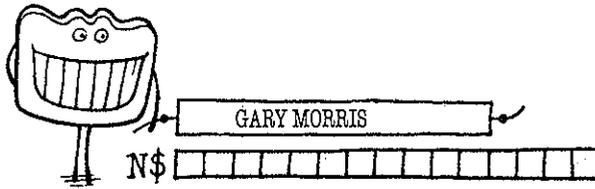
Em seguida ele vai separar um bloco com espaço para quinze caracteres para nele colocar sua resposta.

```
160 DIM N$(15)
```



LOGO VOCÊ VAI SABER
POR QUE PRECISO
FAZER ISTO

Você digita o nome da pessoa cujo número você quer. Ele vai para N\$.



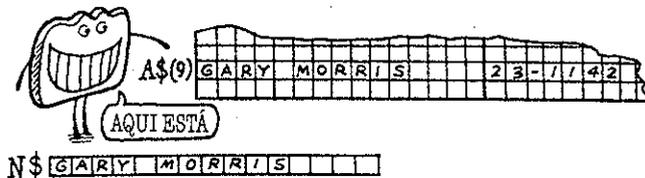
170 INPUT N\$

Depois de limpar a tela, o TK95 vai procurar nos dez registros, olhando os nomes no primeiro campo até encontrar o nome que você digitou. Ele vai então escrever o nome e o número de telefone. Utilizamos um loop FOR...NEXT para fazer com que o computador examine os dez registros, um de cada vez.

```

180 CLS
190 FOR R = 1 TO 10
200 IF A$(R,1)=N$ THEN PRINT AT
    8,0;A$(R,1);AT 10,0;A$(R,2)
210 NEXT R
220 PRINT AT 16,0;"FINAL DA BUS
CA"

```



Você compreende agora porque o nome que você digitou precisava ir para um bloco com 15 compartimentos?



"GARY MORRIS" "ISTO É O
MESMO QUE
"GARY MORRIS", MAS
"GARY MORRIS" É DIFERENTE

Para você fazer

Escreva seu próprio programa de números de telefone usando os nomes de seus amigos.

Talvez você possa também incluir os endereços também. Neste caso você vai precisar de três campos para cada registro, assim:

ANNE SMITH	3 NEW ROAD	709732
PETER MORGAN	101 HIGH STREET	51-0902
JANET MARTIN	15 CHESTNUT AVENUE	304982

O TK95 vai precisar de uma matriz maior para armazenar esta informação. Se você alterar a linha 10 para

```
10 DIM A$(10,3,20)
```

existirá mais espaço para dez registros, cada um com três campos: nome, endereço e número de telefone. Se você colocar 20 como sendo o último número dos parênteses, estará criando espaço para 20 letras em cada campo. Isto deverá ser suficiente para um endereço, mas se um de seus amigos tiver um endereço particularmente longo você vai precisar de um número maior do que 20.

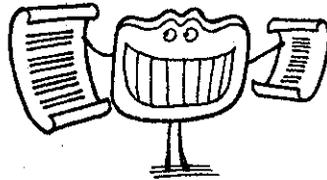


VOCÊ VAI PRECISAR TAMBÉM
ALTERAR
AS LINHAS 30, 150, 160, E
200 E AS LINHAS DATA

8

Para Lá e para Cá

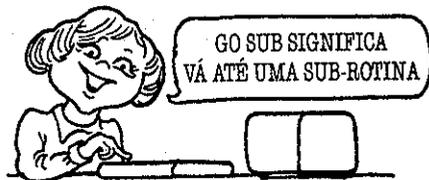
Aqui está um programa em duas partes — a principal e uma outra denominada “sub-rotina”.



```

10 REM Programa principal
20 PRINT "Voce quer saber o que
esta na minha sub-rotina?"
30 INPUT a$
40 IF a$="yes" THEN GO SUB 100
50 IF a$ < > "yes" THEN PRINT AT
10,0;"Amigo da onca!"
60 PRINT AT 20,0:"Este e o fi
nal do programa"
70 STOP

```



```

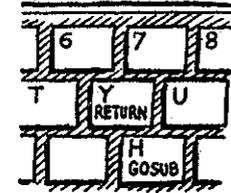
100 REM Sub-rotina
110 FOR c=0 TO 22
120 LET i=INT (RND*7)

```

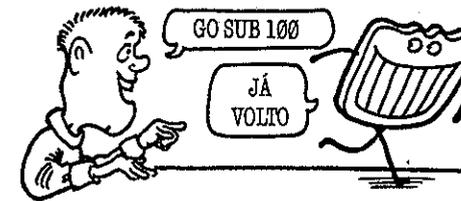
```

130 INK i
140 PRINT AT 10,c;"TK90X"
150 SOUND 15,c
160 NEXT c
170 INK 0:CLS
180 RETURN

```



Na linha 40 do programa principal temos um comando para que o TK95 GO SUB 100. Ele diz ao computador para encontrar a linha 100 e executar a lista de instruções que começa ali. Quando ele encontra a palavra RETURN, sabe que é hora de voltar e continuar com o programa principal.



A palavra STOP, no final do programa principal, é muito importante. O que você acha que aconteceria se a linha 70 ficasse de fora?

Aqui está um programa de perguntas e respostas com duas sub-rotinas.

```

10 REM QUIZ
20 PAPER 7: INK 0: CLS
30 DIM a$(10,2,8)
40 LET score=0
50 REM leia DATA
60 RESTORE 700
70 FOR r=1 TO 10
80 READ a$(r,1),a$(r,2)

```

```

90 NEXT r
100 PRINT AT 2,2; "Coletivos"
110 FOR a=1 TO 10
120 BORDER 3
130 PRINT AT 6,2; "De o coleti
vo'"
140 PRINT AT 8,2;a$(a,1)
150 DIM y$(8): INPUT y$
160 CLS
170 IF y$=a$(a.2) THEN GO SUB
300: REM certo
180 IF y$ < > a$(a,2) THEN GO SUB
500: REM Errado
190 INPUT "Pressione ENTER ";z$
200 CLS
210 NEXT a
220 REM Escore
230 PRINT AT 10,2;"Voce acertou"
; escore;" entre um e 10"
240 STOP

```



```

300 REM Certo
310 LET escore=escore+1
330 PRINT AT 10,2;"Certo"
410 RETURN
500 REM Errado
520 PRINT AT 10,2;"Nao, é?";
a$(a,2)
600 RETURN

```



```

700 DATA "lobos";"alcateia"
710 DATA "camelos";"cafila"
720 DATA "elefantes";"manada"
730 DATA "cachorros";"matilha"
740 DATA "peixes";"cardume"
750 DATA "abelhas";"enxame"
760 DATA "passaros";"revoada"
770 DATA "formigas";"correicao"
780 DATA "carneiros";"rebanho"
790 DATA "gafanhotos";"nuvem"

```

a\$	Campo 1	Campo 2
1	l o b o s	a l c a t e i a
2	c a m e l o s	c a f i l a
3		
4		
5		

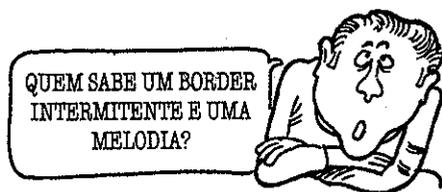
Você encontrará os comandos GO SUB nas linhas 170 e 180 do programa principal.

As perguntas são sobre animais, e seus coletivos, mas se você quiser poderá facilmente alterá-lo e escolher um assunto que lhe agrada. Altere as linhas 100 e 130 e coloque seus próprios dados nas linhas 700 a 790. Quem sabe você gostaria de usar este programa de perguntas e respostas para testar seu vocabulário de inglês?



Para você fazer

Torne a sub-rotina "certo" e "errado" mais divertida, introduzindo som e cor.

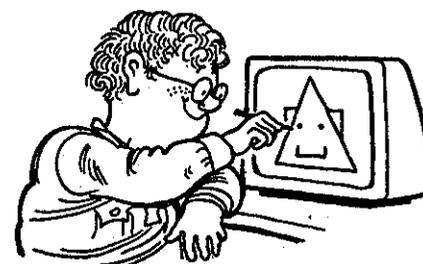


Em outro capítulo mais adiante vamos acrescentar algumas linhas a este programa de perguntas e respostas. Se você gravar o programa numa fita não precisará digitá-lo outra vez.

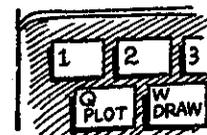
9

Marque um Ponto e Trace uma Linha

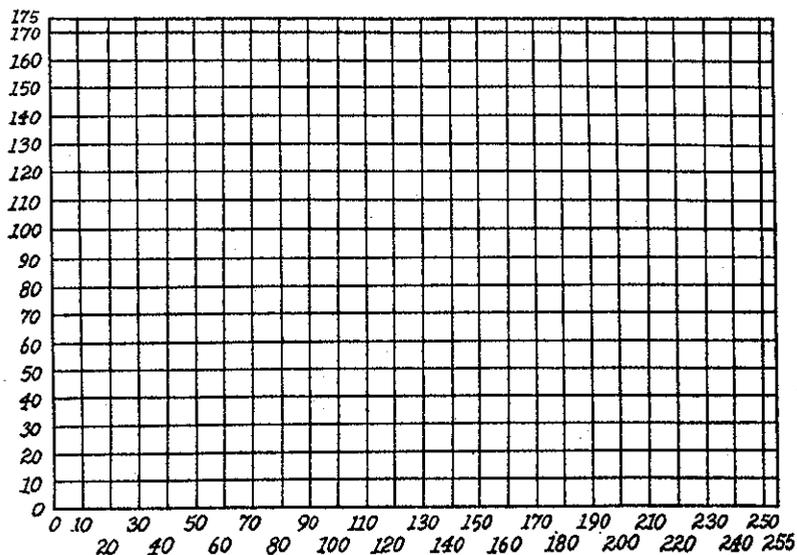
No seu TK95 você pode usar gráficos de "alta resolução", isto significa que você pode criar na tela figuras e diagramas precisos.



Em BASIC as palavras utilizadas para traçar linhas são PLOT e DRAW. Você vai encontrá-las escritas em branco próximo à parte esquerda superior do teclado. Quando você precisar de uma delas basta pressioná-la.



Imagine sua tela dividida em quadrados como estes:

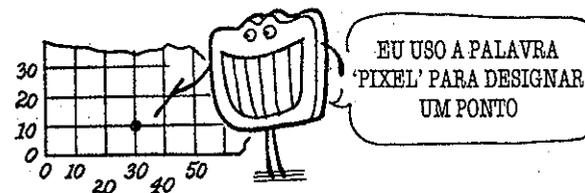


Os números na horizontal começam em 0 no canto inferior esquerdo e terminam com 255 à direita. A linha que contém estes números é normalmente denominada "eixo de x".

Quando você quiser que o TK95 desenhe um ponto na tela, é preciso dizer a ele a que distância do centro, e a que altura o ponto deve ficar. Digite:

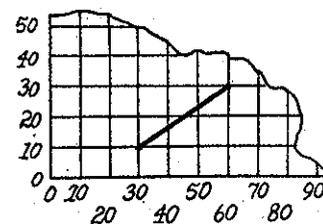
```
PLOT 30,10
```

e um ponto aparecerá próximo ao canto esquerdo inferior da tela.



30 e 10 são as "coordenadas" do ponto ou elemento de imagem.

Quando você tiver marcado um ponto use a palavra DRAW para traçar uma linha a partir dele. A palavra DRAW deve ser seguida por dois números que dizem ao TK95 até onde deve traçar. Suponhamos que você quer esta linha



De onde estamos, o ponto para onde estamos traçando está 30 espaços para a direita e 20 para cima. Digite

```
DRAW 30,20
```



Se quisermos que o TK95 trace para a esquerda devemos colocar um sinal de menos antes do número na horizontal. Experimente isto:

```
DRAW -40,10
```

Para traçar para baixo colocamos um sinal de menos diante do número na vertical. Diga ao computador para

```
DRAW 10,-30
```



Este é um programa para desenhar um quadrado:

```
10 PLOT 100,60
20 DRAW 50,0
30 DRAW 0,50
40 DRAW -50,0
50 DRAW 0,-50
```

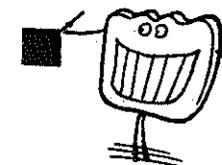


Este quadrado contém um outro:

```
60 PLOT 110,70
70 DRAW 30,0
80 DRAW 0,30
90 DRAW -30,0
100 DRAW 0,-30
```

O TK95 não tem um comando especial para colorir ou preencher uma forma, mas você pode dar a um quadrado uma aparência de sólido desenhando várias linhas, uma ao lado da outra, assim:

```
10 FOR y=55 TO 75
20 PLOT 60,y
30 DRAW 20,0
40 NEXT y
```



Um triângulo sólido é um pouco mais complicado:

```
10 LET s=20
20 FOR y=50 TO 70
30 PLOT 60,y
40 DRAW s,0
50 LET s=s-1
60 NEXT y
```



Você pode, é claro, traçar linhas coloridas, dizendo ao computador qual a cor para INK e alterar a cor do fundo com PAPER. Inclua esta linha no programa para o triângulo e execute-o outra vez.

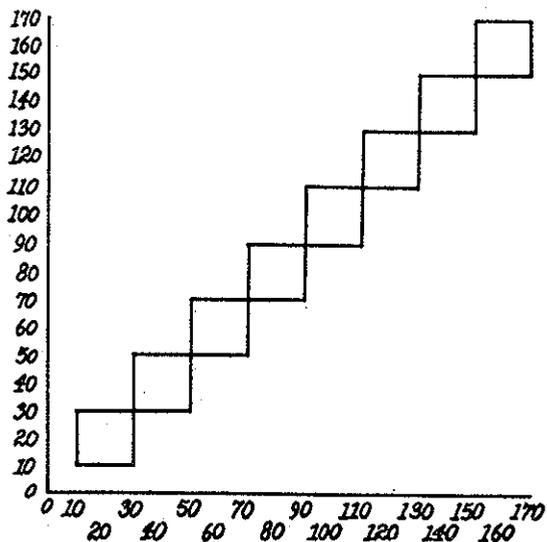
```
5 PAPER 0: INK 5:CLS
```

Para você fazer

Experimente se você é capaz de desenhar estas formas na tela. Escolha as cores.



FAÇA ESTE
DESENHO USANDO
DOIS TRIÂNGULOS



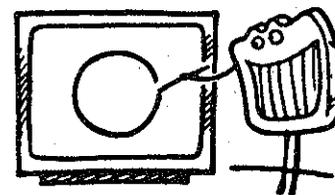
Você vai precisar de um loop FOR...NEXT para a cadeia de quadrados.

10 Círculos

Embaixo da tecla H, você vai encontrar, escrita em vermelho, a palavra CIRCLE. Quando você quiser que o TK95 desenhe um círculo na tela será preciso que você indique onde deve ficar o centro deste círculo e o comprimento do raio. Digite

```
CIRCLE 130,90,50
```

e você terá um círculo próximo ao centro da tela.



130 e 90 são as coordenadas do centro do círculo. 50 é o raio. Tente criar seus próprios círculos. Você talvez ache mais fácil decidir onde deve ficar o centro do círculo e qual o comprimento do raio, olhando o quadriculado da página 44.

CUIDADO PARA QUE
SEUS CÍRCULOS NÃO
SAIAM DA TELA!



Este é um programa curto que traça círculos concêntricos. Todos possuem o mesmo centro (120,80) mas os raios são diferentes.

```
10 FOR r=5 TO 60 STEP 5
20 CIRCLE 120,80,r
30 NEXT r
```

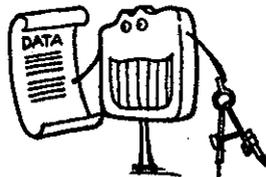
Agora, vamos desenhar dez círculos cada um com raio 20. O TK95 é quem vai decidir onde colocá-los na tela.

```
10 FOR c=1 TO 10
20 LET x=INT (RND*236)+20
30 LET y=INT (RND*156)+20
40 CIRCLE x,y,20
50 NEXT c
```



Se você quiser desenhar vários círculos, pode colocar as coordenadas e os raios em uma lista de DADOS como esta:

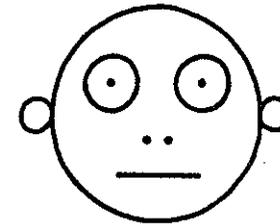
```
10 FOR c=1 TO 9
20 READ x,y,r
30 CIRCLE x,y,r
40 NEXT c
50 DATA 130,90,50,80,90,20,180,90,20
```



```
60 DATA 130,40,20,130,140,20,95,55,30
70 DATA 95,125,30,165,125,30,165,55,30
```

Para você fazer

1) Faça um programa para desenhar este rosto:



As pupilas e as narinas são pontos simples.

2) Aqui está um programa no qual o TK95 pergunta se você gostaria de ver quadrados e círculos coloridos. Tem 2 sub-rotinas — quadrados e círculos. A sub-rotina dos quadrados já está feita mas você terá que criar uma para os círculos.

```
10 REM Formas
20 INK 0: BORDER 7
30 PRINT "FORMAS"
40 PRINT AT 4,0:"Eu sei desenha
r"
50 PRINT AT 6,0:"circulos"
60 PRINT AT 7,0:"quadrados"
70 PRINT AT 15,0:"Qual deles v
oce quer?"
80 PRINT AT 18,0:"Para terminar di
gite stop"
90 INPUT a$
```

```

100 CLS
110 IF a$="circulos" THEN GO SUB
200
120 IF a$="quadrados" THEN GO S
UB 300
130 IF a$="stop" THEN STOP
140 RUN
200 REM circulos
280 INPUT "Pressione ENTER",b$
290 RETURN
300 REM quadrados
310 FOR a=1 TO 40
320 LET c=INT (RND*7)
330 LET x=INT (RND*200)
340 LET y=INT (RND*125)
350 LET n=INT (RND*40)+10
360 PLOT x,y
370 SOUND .2,n
380 BORDER c: INK c
390 DRAW n,0: DRAW 0,n: DRAW -n
0: DRAW 0, -n
400 NEXT a
410 INPUT "Pressione ENTER" .b$
420 RETURN

```



11

O TK95 Indica um Ponto

Se você escolher um determinado ponto da tela, o TK95 pode dizer se ali já existe um elemento de imagem.



Limpe a tela usando CLS e ENTER e vamos pedir ao computador que procure na posição indicada pelas coordenadas 100,150. Digite

```
PRINT POINT (100,150)
```



Se ali existe um ponto marcado, o TK95 responderá com o número 1. Se não, o computador responderá com um 0. Limpe a tela outra vez e crie um ponto com

PLOT 100,150

Agora peça ao TK95 para

PRINT POINT (100,150)



Aqui está um programa onde o TK95 traça uma linha de pontos na tela de 200,0 até 200,175. Ele então vai traçar uma segunda linha de pontos de um lado a outro da tela, de 0,100 a 255,100. A segunda linha cruza a primeira.

```
10 FOR y=0 TO 175
20 PLOT 200,y
30 NEXT y
40 FOR x=0 TO 255
60 PLOT x,100
70 NEXT x
```

Rode o programa e acrescente então a linha 50:

```
50 IF POINT (x,100)=1 THEN STOP
```

Quando você rodar novamente o programa o que acontece é o seguinte: antes de marcar cada ponto através da tela, o computador verifica se aquela posição já foi marcada antes. Se já tiver sido marcada o programa será interrompido.



O próximo programa chama-se "Garatujas". Assim que você começa a rodar o programa o TK95 começa a traçar uma linha de pontos através da tela, movimentando-se para a direita. Para mudar a direção do movimento, pressione

- a — para cima
- z — para baixo
- n — para a esquerda
- m — para a direita

Não é necessário manter sempre a tecla pressionada, basta pressionar quando quiser mudar de direção.

CUIDADO PARA NÃO ESBARRAR NA BORDA POIS O PROGRAMA SERÁ INTERROMPIDO



```
10 REM Garatujas
20 BORDER 4: INK 2: CLS
30 LET x=0: LET y=90
40 LET d$="m"
```

d\$ É A DIREÇÃO DO MOVIMENTO DA LINHA. "m" MOVIMENTO PARA A DIREITA



```

50 PLOT x,y
60 LET i$=INKEY$
70 IF i$ < > "" THEN LET d$=i$
80 IF d$="a" THEN LET mx=0: L
ET my=1
90 IF d$="z" THEN LET mx=0: L
ET my=-1
100 IF d$="m" THEN LET mx=1: L
ET my=0
110 IF d$="n" THEN LET mx=-1:
LET my=0
120 LET x=x+mx: LET y=y+my
130 IF x < 0 OR x > 255 THEN GO TO
170
140 IF y < 0 OR y > 175 THEN GO TO
170
160 GO TO 50
170 INK 1: PRINT AT 10,10;"Co
lisao!"
180 SOUND 1,7

```

Se você tiver uma impressora ZX vai poder fazer uma cópia das suas garatujas em papel. O próximo capítulo ensina a executar esta tarefa. Talvez agora você queira salvar o programa em fita assim ele estará pronto para ser usado outra vez com a impressora.

Para você fazer

Acrescente uma linha ao programa garatuja (linha 150), para que o TK95 passe para a linha 170 se você tentar atravessar uma linha que já tenha sido desenhada.



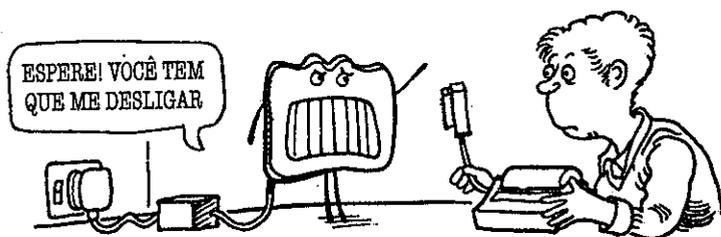
12

A sua Impressora ZX

Se você possui uma impressora ZX poderá imprimir e conservar por escrito todos os seus programas.



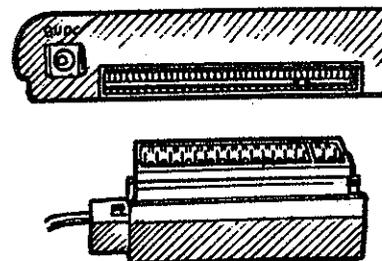
Primeiro é preciso acoplar a impressora ao seu TK95. É muito importante que o TK95 não esteja ligado enquanto você faz esta conexão pois isto poderia danificar tanto a impressora quanto o computador.



Na parte de trás do TK95 existe uma abertura na cobertura plástica. Através dela você pode ver a borda do quadro ao qual estão fixados todos os componentes do computador.

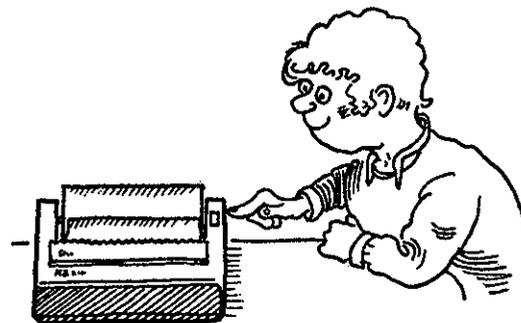


É este o local para a conexão da impressora. Pressione a tomada azul da impressora contra a borda do quadro no computador assegurando-se de que a listinha branca na tomada de conexão da impressora esteja em alinhamento com a abertura no quadro.



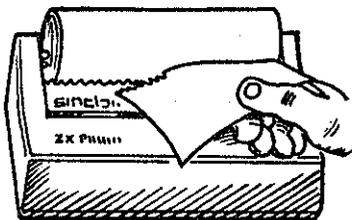
Agora você pode ligar a energia.

Quando você recebe a sua impressora, ela vem carregada com um rolo de papel no qual está impresso um desenho circular. Para retirar este desenho aperte um pequeno botão preto que está localizado à direita da impressora e ele fará sair o papel.



Tire o dedo do botão e rasgue o papel. A impressora possui um dispositivo serrilhado para esta finalidade.

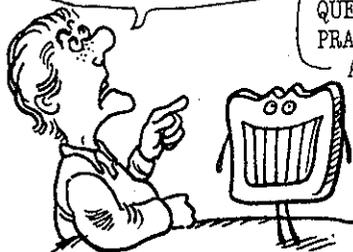
É MAIS FÁCIL CORTAR
O PAPEL NESTE ÂNGULO



Existem três comandos em BASIC para serem usados com a impressora — COPY, LPRIN e LLIST. Todos estão na carreira inferior do teclado. COPY está escrita em branco e as outras em verde.

Se você tem um programa na memória do TK95 mude para o cursor E e pressione LLIST e em seguida ENTER. O programa será impresso no papel. A impressora provavelmente vai lhe dar um susto quando começar a funcionar. Ao contrário do TK95, ela não trabalha silenciosamente.

O QUE SERÁ ESTE CHEIRO
DE QUEIMADO?



A IMPRESSORA FUNCIONA
QUEIMANDO O REVESTIMENTO
PRATEADO DO PAPEL PARA QUE
APAREÇA A PARTE PRETA
QUE ESTÁ POR BAIXO

Com COPY, você poderá copiar no papel qualquer coisa que esteja na tela naquele momento. Tente executar este programa e veja o que o TK95 vai desenhar na tela. Quando ele tiver acabado — vai demorar bastante — pressione COPY e ENTER. A impressora passará o desenho para o papel.

```
10 FOR x=-100 TO 100
20 LET y=x*x
30 CIRCLE x+120,y/100+20,20
40 NEXT x
```



ERA ISTO MESMO QUE
EU PRECISAVA PARA
REGAR O MEU JARDIM

Você também pode colocar COPY num programa. Coloque "Garatujas" no seu computador e acrescente esta linha:

```
115 IF d$="c" THEN COPY
```

Rode o programa. Cada vez que você pressiona "c" a impressora copiará a tela. Aqui está um exemplo:

```
10 LPRINT "Digite seu nome,por  
favor"  
20 INPUT n$  
30 FOR a =1 TO 50  
40 LPRINT n$;  
50 NEXT a
```

Você não pode usar LPRINT AT. Se você pensar bem, logo vai descobrir porque.



Entretanto você pode usar a palavra TAB, para dizer ao computador em que coluna deve imprimir. TAB pode ser usada tanto com PRINT quanto com LPRINT.



Estas são algumas linhas adicionais para serem acrescentadas ao programa de perguntas e respostas do Capítulo 8. Elas substituem a seção `escore` (linhas 220 a 240), e darão a você um diploma provando como você ganhou bem o jogo.

```

220 REM Diploma
230 INPUT "Digite seu nome, por
favor";n$
240 LPRINT "*****
*****"

245 LPRINT
250 LPRINT TAB 4;"Coletivos"
255 LPRINT
260 LPRINT TAB 4;n$

```

```

265 LPRINT
270 LPRINT TAB4; "Voce acertou"
;escore; "de 1 a 10"
275 LPRINT
280 LPRINT "*****
*****"
290 STOP

```

13

Caracteres e Códigos

Cada uma das palavras e cada um dos caracteres que você pode digitar no teclado do TK95 é acompanhado por um número chamado número de Código (CODE number). O A por exemplo tem como número de código 65. Para que o TK95 diga isto, digite:

```
PRINT CODE "A"
```

Se você der um número ao TK95 ele poderá dizer qual o caractere ou palavra que o acompanha.

```
PRINT CHR$ 245
```

fará com que ele responda PRINT, porque o número de Código de PRINT é 245.



Digite este programa e o computador vai apresentar o número de Código de cada palavra e de cada caractere do teclado:

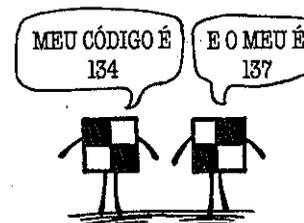
```
10 FOR c=32 TO 255  
20 PRINT c, CHR$ c
```

```
30 PRINT  
40 NEXT c
```

Quando ele perguntar "scroll" (rolar a tela)? Pressione qualquer tecla (exceto N ou BREAK) e ele apresentará a tela seguinte. No manual do seu TK95 você encontrará uma lista de caracteres e seus códigos.



Talvez você já tenha percebido que os códigos de 126 a 143 pertencem aos caracteres gráficos — aqueles que estão na carreira superior do teclado e para os quais você precisa do cursor G.



Os próximos vinte e um números de código, de 144 a 164, parecem pertencer a um segundo conjunto de letras maiúsculas de A a V. Mas na verdade eles são números de código extra e você pode escolher seus próprios caracteres gráficos para eles. Isto será explicado no próximo capítulo.

Para você fazer:

Você é capaz de decifrar esta mensagem?

```
PRINT CHR$ 86;CHR$ 111;CHR$ 99  
;CHR$ 101;"APRENDEU";CHR$ 111  
;CHR$ 115;CHR$ 67;CHR$ 111;CHR$  
100;CHR$ 105;CHR$ 103;CHR$ 11  
1;CHR$ 115
```

14

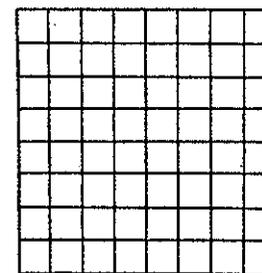
Seus Próprios Caracteres

Os jogos ficam muito mais divertidos no seu TK95 quando envolvem pessoas reais, carros, foguetes etc. Você pode escolher seus próprios caracteres e colocá-los na memória do TK95, prontos para serem usados num programa.

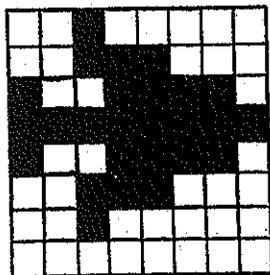


Os caracteres especiais projetados por você são denominados "símbolos gráficos definidos pelo usuário".

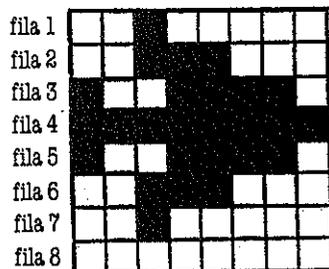
Em primeiro lugar você precisa desenhar em papel quadriculado, um quadrado com oito espaços de lado. Quando seu caractere aparecer na tela ocupará um espaço correspondente a 8 x 8 elementos.



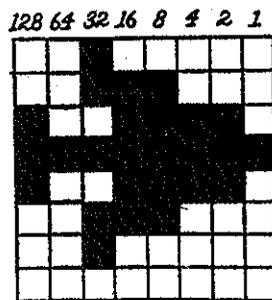
Agora desenhe seu próprio caractere sombreando alguns dos quadrados. Isto é um avião visto de cima.



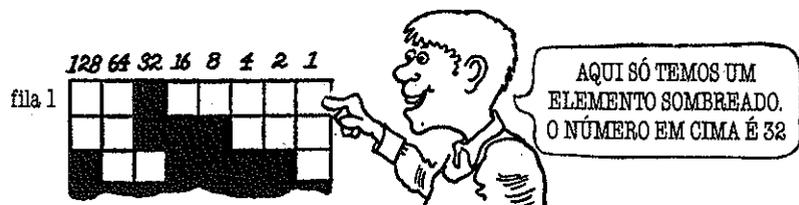
Numere as linhas de seu quadriculado assim:



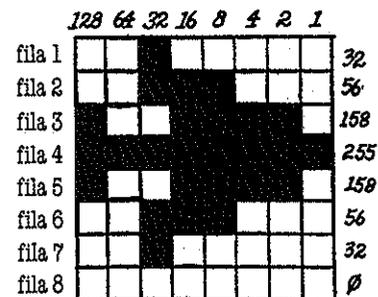
e escreva estes números na parte de cima das colunas:



Observe agora a linha 1 de seu quadriculado. Sempre que você encontrar um quadrado sombreado anote o número que o acompanha. Quando terminar, some os números e anote o total.



Faça o mesmo com as outras linhas de forma que ao terminar você terá oito números, um para cada linha.

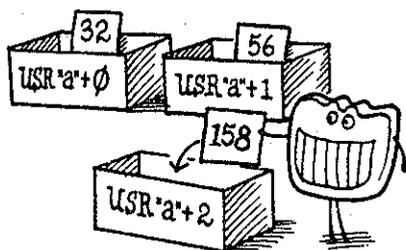


No capítulo anterior vimos que o TK95 tem espaço em sua memória para 21 caracteres novos e existe um número de código reservado para cada um deles. CODE 144 pertence temporariamente a um A de reserva. Faremos uma alteração para que ele passe a pertencer ao nosso avião. O TK95 usará 8 unidades de memória para armazenar a informação sobre o caractere — uma para cada linha do quadriculado. Este programa vai colocar o avião na memória do computador:

```

10 FOR a=0 TO 7
20 READ n
30 POKE USR "a"+a,n
40 NEXT a
50 DATA 32,56,158,255,158,56,3
2,0

```

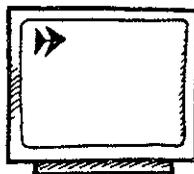


POKE significa 'coloque em'. É preciso dizer ao TK95 *onde* colocar e então, depois da vírgula, *o quê* colocar.

Rode o programa. Para ver seu caractere aparecer na tela, digite:

```
PRINT "
```

Mude agora para o cursor G e pressione A. Volte ao cursor L e termine com uma outra série de aspas. Pressione ENTER.



Esta é uma outra maneira de fazer surgir o avião:

```
PRINT CHR$ 144
```

Uma vez que você tenha rodado este programa o avião estará arquivado na memória do TK95 e pressionando A com o cursor G você fará com que ele surja na tela. Mesmo que você pressione NEW, ele continuará lá até que você desligue o computador ou até que você desenhe outro caractere para a tecla de caracteres gráficos A.

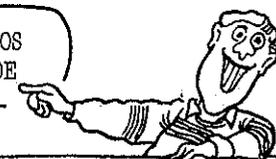
Aqui está um programa no qual o avião atravessa a tela. Para explodí-lo em pleno vôo, pressione a tecla P.

```

10 FOR a=0 TO 7
20 READ n
30 POKE USR CHR$ 144+a,n
40 NEXT a
50 DATA 32,56,158,255,158,56,3
2,0

```

NA LINHA 30 USAMOS
CHR\$ 144 EM VEZ DE
"a"



```

60 PAPER 5:CLS
70 INK 2
80 FOR c=0 TO 30
90 PRINT AT 10,c;" A"
100 IF INKEY$ ="p" THEN GOTO 13
0
110 NEXT c
120 STOP

```

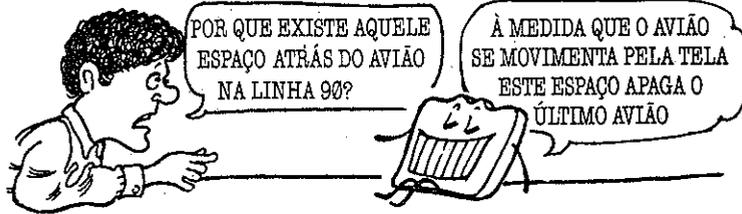


VOCÊ DEVE MUDAR O CURSOR
PARA G
ANTES DE DIGITAR A NO FINAL
DA
LINHA 90.
DEPOIS MUDE OUTRA VEZ PARA L

```

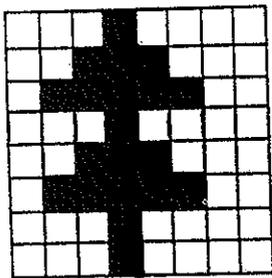
130 PRINT AT 10,c;"*"
140 FOR b=24 TO 0 STEP-1
150 SOUND .01,b
160 NEXT b
170 PRINT AT 10,c;" "

```

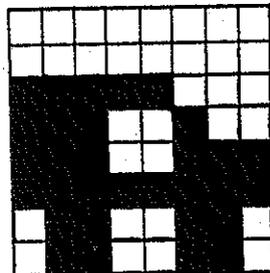


Para você fazer:

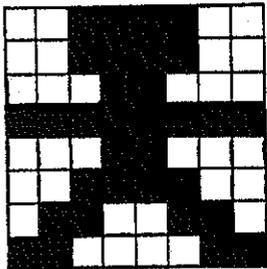
Talvez você queira usar alguns dos caracteres abaixo em seus programas.



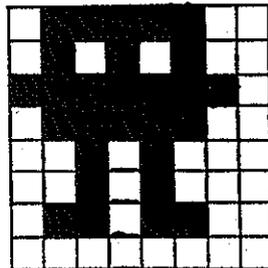
Árvore



Furgão



Homem



Invasor

Calcule os oito números para cada um deles e coloque-os nas linhas 70 a 100 deste programa:

```

10 FOR c=144 TO 147
20 FOR a=0 TO 7
30 READ n
40 POKE USR CHR$ c+a,n
50 NEXT a
60 NEXT c
70 DATA
80 DATA
90 DATA
100 DATA

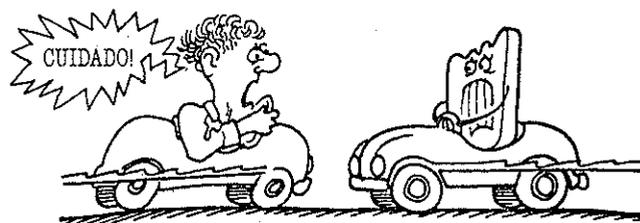
```

Os quatro novos caracteres terão como números de código 144, 145, 146 e 147. Você pode colocá-los na tela com o cursor G e A, B, C e D.

15

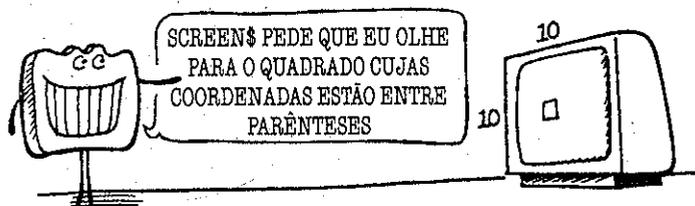
Esbarrei em Alguma Coisa?

Em qualquer dos programas onde os objetos se movem ao redor da tela é muito útil saber se há risco de colisão.



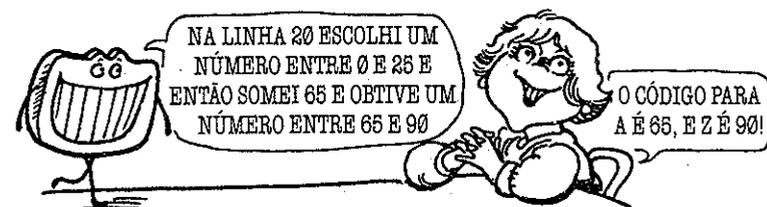
Se você disser ao computador o número da linha e os números da coluna referentes a uma posição na tela, o TK95 pode examinar o quadrado e dizer o que existe naquele determinado local. A palavra em BASIC que deve ser usada é SCREEN, que está escrita em verde embaixo da tecla K. Neste programa você poderá ver como ela funciona:

```
10 PRINT AT 10,10;"A"  
20 PRINT AT 15,0;"Em 10,10 Veja";  
30 PRINT SCREEN$(10,10)
```



SCREEN\$ é particularmente útil em programas onde o TK95 escolhe ao acaso onde vai fazer aparecer os objetos. No próximo programa, o computador preenche as dez primeiras linhas da tela com as letras maiúsculas que escolher. Ele pede que você dê o número de uma linha e o número de uma coluna e em seguida diz o que existe ali.

```
10 FOR c=1 TO 320  
20 LET r=INT (RND*26)+65  
30 PRINT CHR$ r;  
40 NEXT c  
50 INPUT "Diga o numero de uma  
linha (0 a 9)"; linha  
60 INPUT "Diga o numero de uma  
coluna" "" (0 a 31)"; coluna  
70 PRINT : PRINT  
80 PRINT "Na "; linha;" "; colun  
a:" Vejo ";  
90 PRINT SCREEN$ (linha,coluna)
```



Com SCREEN\$, o TK95 só pode ver letras, números e os símbolos em vermelho em cima das teclas; os caracteres gráficos para os quais usamos o cursor G não podem ser percebidos por ele.



Experimente isto:

```
10 PRINT AT 0,0;"■"  
20 PRINT AT 10,0;"Em 0,0 Vejo";  
30 PRINT SCREEN$(0,0)
```

No Capítulo 17 você encontrará um meio de verificar estes caracteres gráficos.

O próximo programa chama-se "Para-quedista". Um homem está tentando chegar ao solo mas naquela área existem muitos arranha-céus. Use a tecla 5 para movimentá-lo para a esquerda, e a 8 para a direita. O computador vai usar SCREEN\$ para fazê-lo parar se esbarrar num prédio. Você vai precisar do cursor Graphics para o A na linha 200.

```
10 REM Para-quedista  
20 FOR a=0 TO 7  
30 READ n  
40 POKE USR "a"+a,n  
50 NEXT a  
60 DATA 56,124,124,68,84,56,16  
40
```



```
70 BORDER 5: PAPER 7: CLS  
80 INK 0  
90 FOR c=0 TO 31  
100 LET s=INT (RND*5)  
110 IF s=0 THEN GO TO 160
```

```
120 LET b=INT (RND*10)+10  
130 FOR a=b TO 21  
140 PRINT AT a,c;"X"  
150 NEXT a  
160 NEXT c
```

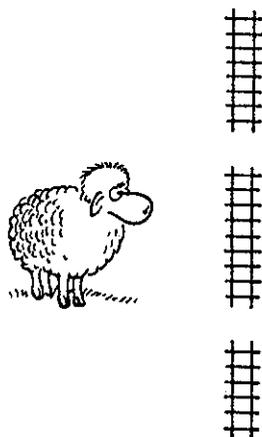


```
170 LET mc=INT (RND*32)  
180 INK 2  
190 FOR r=0 TO 21  
200 PRINT AT r,mc;"A"  
210 SOUND .05,24  
220 IF SCREEN$(r+1,mc)="X" THE  
N PRINT AT 6,14;"SOCORRO":STOP  
230 IF r < 21 THEN PRINT AT r,mc  
;" "  
240 LET d$=INKEY$  
250 IF d$="5" AND m > 0 THEN LE  
T mc=mc-1  
260 IF d$="8" AND mc < 31 THEN L  
ET mc=mc+1  
270 NEXT r
```



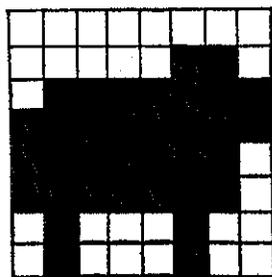
Para você fazer:

Aqui está uma sugestão para um programa: Um carneirinho está preso em um prado e quer sair. Entretanto, existem apenas algumas aberturas na cerca e ele não é muito inteligente, anda sempre em linha reta.



Você deve fazer com que ele ande pelo prado e pare se esbarrar na cerca.

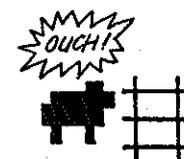
O programa é dividido em três partes principais. Em primeiro lugar você vai precisar desenhar o carneiro e colocá-lo na memória do seu TK95. Apresentamos abaixo uma sugestão para o carneiro:



Em seguida você vai desenhar a cerca com aberturas colocadas ao acaso. Use o símbolo # na tecla 3 para fazer a cerca. Você pode fazer isto assim.

```
90 FOR f=0 TO 21
100 LET r=INT (RND*3)
110 IF r=0 THEN PRINT AT f,20
;""
120 IF r< >0 THEN PRINT AT f,20;
""#
130 NEXT f
```

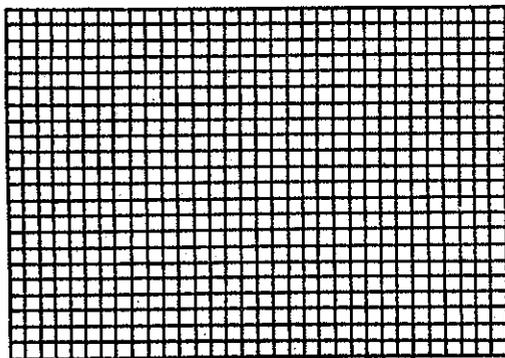
Agora faça o carneirinho andar pelo prado verificando sempre se existe uma parte da cerca no caminho.



16

Efeitos Especiais

Como você sabe, a tela do TK95 tem setecentos e quatro espaços nos quais ele pode imprimir um caractere.



TRINTA E DUAS COLUNAS,
VINTE E DUAS LINHAS.
UM TOTAL DE SETECENTOS
E QUATRO QUADRADOS!

Você pode escolher uma cor de papel e uma cor de tinta para cada quadrado. Este programa determina as cores para oito quadrados:

```
10 PAPER 4; INK 2
20 PRINT AT 10,9;"TK95"
```

Rode o programa e então pressione ENTER para ver outra vez a sua listagem. A linha 10 do programa alterou as cores de INK e PAPER e sua listagem é também vermelho sobre verde:

PREFIRO PRETO
SOBRE BRANCO



Agora pressione NEW para limpar a memória do computador. Apresentamos abaixo uma versão diferente do último programa. Digite e rode.

```
10 PRINT PAPER 4; INK 2; AT 10,9
;"TK 95"
```



PARECE IGUAL A
OUTRA VEZ

Pressione ENTER. Sua listagem ainda apresenta preto sobre branco. Porque os comandos PAPER e INK foram colocados depois da palavra PRINT, seu efeito foi apenas temporário. Alteram somente as cores na linha em que se encontram. Acrescente a linha 20:

```
20 PRINT AT 12,9;"COMPUTADOR"
```

Rode outra vez o programa e verá que os comandos para cores não são válidos para esta linha.

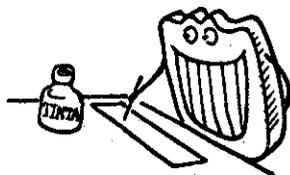


AO USAR OS COMANDOS
INK E PAPER COM PRINT
LEMBRE-SE DOS ;

Na última fileira do teclado, depois de INK e PAPER, você vai encontrar as palavras FLASH, BRIGHT, OVER e INVERSE. Estas são palavras que servem para efeitos especiais e você pode usá-las para realçar o que foi impresso na tela.

Em primeiro lugar FLASH. Digite

```
10 PRINT INK 1;PAPER 6;"azul  
sobre fundo amarelo"
```



Quando você rodar este programinha vai ser exatamente o que ele diz — tinta azul sobre papel amarelo. Agora altere a linha para que fique assim:

```
10 PRINT INK 1; PAPER 6; FLASH  
1;"azul sobre fundo amarelo"
```



FLASH 1 é o comando para tornar a imagem intermitente. Como você pode observar, ao rodar o programa as cores de "ink" e "paper" se alternam. Tente escrever um programa que coloque seu nome na tela usando as cores azul e ciano.

Será que faz alguma diferença se você permutar as cores de INK e PAPER do programa?

FLASH 0 anula o comando FLASH 1. Você pode observar isto no próximo programa.

```
10 FLASH 1  
20 FOR f=1 TO 60  
30 LET i=INT (RND*8)  
40 LET p=INT (RND*8)  
50 PRINT INK i; PAPER p;FLAS  
HING";  
60 NEXT f  
70 FLASH 0  
80 PRINT AT 18,0;"Rode novamente  
e vera que as cores estao difere  
ntes"
```



BRIGHT é útil para realçar palavras na tela. Tente isto:

```
10 INPUT BRIGHT 1; "Por favor di  
gite seu nome ";n$
```



Faça brilhar usando BRIGHT 1 e apague com BRIGHT 0. Se você quiser que toda a tela brilhe digite

BRIGHT 1: CLS

O que fazer para que tudo volte ao normal?



PARA DESTACAR BEM QUALQUER COISA VOCÊ PODE USAR FLASH E BRIGHT JUNTOS

Para destacar palavras INVERSE também é muito bom. No caso de um jogo que tenha uma folha de instruções no início você pode segui-las com

```
100 PRINT INVERSE 1; AT 20,6;"Di
gite ENTER para continuar"
```



COMO VOCÊ CONSEGUIU COLOCAR AS PALAVRAS EXATAMENTE NO MEIO DA LINHA?



'PRESS ENTER TO GO ON' TEM VINTE CARACTERES. UMA LINHA TEM 32. PORTANTO SOBRAM 12... COLOCAMOS 6 NO INÍCIO E 6 NO FINAL

INVERSE quer dizer "o oposto". Na verdade o que o computador faz é permutar as cores de INK e PAPER. Use INVERSE 1 para inverter e INVERSE 0 para voltar ao normal.

Espero que você saiba o que este programa faz:

```
10 PRINT AT 0,0;"OOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOO"
20 PAUSE 50
30 PRINT AT 0,0;"XXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXX"
```

Quando você tiver rodado o programa altere a linha 30 de modo que a listagem agora seja:

```
10 PRINT AT 0,0;"OOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOO"
20 PAUSE 50
30 PRINT OVER 1; AT 0,0;"XXXXX
XXXXXXXXXXXXXXXXXXXXX"
```

Rode outra vez o programa.

NUNCA VI CARACTERES COMO ESTES!



O TK95 colocou X na parte de cima dos O. É interessante saber a respeito de OVER que se você disser ao TK95 para colocar uma outra linha de X sobre a nossa linha de caracteres misturados poderemos na verdade fazer com que ele apague os X. Acrescente estas duas linhas ao programa:

```
40 PAUSE 50
50 PRINT OVER 1; AT 0,0;"XXXXX
XXXXXXXXXXXXXXXXXXXXX"
```

Será que você consegue alterar a linha 50 para que em vez de apagar os X ele apague os O?

OVER é muito útil quando você quer que um objeto se movimente através da tela sem apagar o fundo. Aqui está uma nave que passa pelo sol:

```
10 BORDER 5; PAPER 7: INK 6: CLS
20 FOR a=5 TO 25 STEP 5
30 CIRCLE 100,100,a
40 NEXT a
50 FOR c=0 TO 31
60 PRINT INK 0; OVER 1:AT 10,
c;"+"
70 PAUSE 10
80 PRINT OVER 1;AT 10,c;"+"
90 NEXT c
100 INK 0
```

Para você fazer:

Acrescente uma linha ao programa do sol para que a tela fique brilhante. Faça com que a nave pisque ao atravessar a tela.

17

Atributos

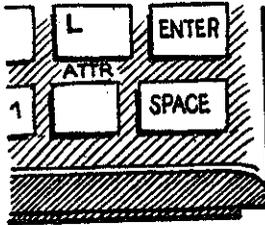
No Capítulo 15 verificamos que ao usar SCREEN\$, podemos fazer com que o computador examine um quadrado na tela e diga o que existe nele. Entretanto, ele não pode reconhecer os caracteres gráficos. Ainda bem que existe um outro método para fazer com que o TK95 examine as posições na tela e desta vez, isto serve para todos os caracteres.



O TK95 pode examinar um quadrado e descobrir:

Se está fixo ou intermitente,
Se está brilhante ou normal,
A cor para PAPER,
A cor para INK.

Estes quatro itens são denominados 'atributos', neste caso daquele quadrado, e a palavra em BASIC para fazer com que o TK95 procure por eles é ATTR.



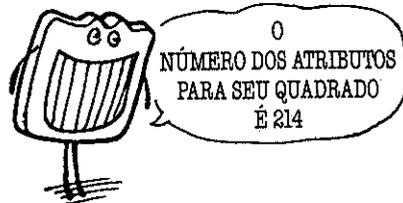
Se você disser ao computador para

```
PRINT ATTR(10,20)
```

ele vai apresentar um número, descrevendo os atributos do quadrado que está na linha 10, coluna 20.

Vamos tentar este programa:

```
10 PRINT FLASH 1; BRIGTH 1; P  
APER 2; INK 6; AT 10,20; " " "  
20 PRINT ATTR (10,20)
```



Para chegar ao número dos atributos o TK95 soma quatro números:

- | | |
|--|-------------------|
| (1) 128 se o caractere estiver intermitente: | 0 se estiver fixo |
| (2) 64 se estiver brilhante: | 0 se não estiver |
| (3) 8 vezes a cor de PAPER | |
| (4) A cor de INK | |

Portanto para nosso caractere intermitente, brilhante, vermelho e amarelo ele somou

- 128 intermitente
- 64 brilhante
- 16 8 vezes a cor de PAPER
- 6 a cor de INK

214

Altere o programa de modo que o caractere esteja normal em vez de brilhante. Tente calcular o número dos atributos antes de rodar o programa.

Aqui está um programa no qual um elefante tenta encontrar seu caminho na floresta. Depois de programar os caracteres gráficos para a árvore e para o elefante, o computador apresentará 50 árvores na tela, ao acaso (linhas 130 a 170). Quando ele perguntar "Qual a linha para o elefante?", digite um número entre 0 e 21 e então o elefante vai andar pela tela. ATTR na linha 220 verifica se ele encontrou alguma árvore.

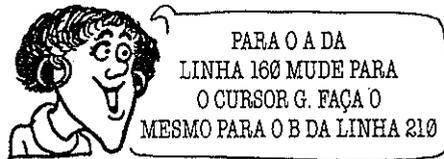


- ```
10 REM elefante
20 REM Caracteres Gráficos
30 FOR c=144 TO 145
40 FOR a=0 TO 7
50 READ n
60 POKE USR CHR$ c+a,n
70 NEXT a
```

```

80 NEXT c
90 DATA 16,56,124,16,56,124,16
16
100 DATA 0,0,126,251,255,125,69
68
110 REM fundo
120 BRIGHT 1: BORDER 0: PAPER 0
CLS
130 FOR t=1 TO 50
140 LET linha=INT (RND*22)
150 LET coluna=INT (RND*22)+10
160 PRINT INK 4;AT linha,coluna;
"A"

```



```

170 NEXT t
180 REM O elefante anda
190 INPUT "Qual a linha para o
elefante? ";e
200 FOR c=0 TO 31
210 PRINT BRIGHT 0, INK 7;AT e
c;"B"
220 IF ATTR (e,c+1)=68 THEN SO
UND 2,-10: STOP
230 PAUSE 10
240 PRINT AT e,c;" "
250 NEXT c
260 FOR b=0 TO 12
270 SOUND .1,b
280 NEXT b

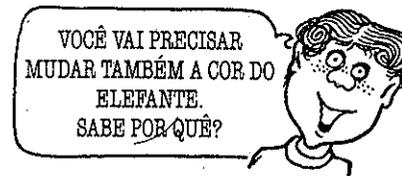
```

Para você fazer:

1. Altere o programa do elefante de modo que:

A tela fique normal em vez de brilhante  
BORDER seja verde e PAPER branco

Deixe que as árvores continuem verdes. O número de atributos na linha 220 deverá ser alterado porque as árvores não serão mais brilhantes e a cor de PAPER que corresponde a elas será diferente.



2. Estas são algumas outras alterações que você poderá fazer:

Na linha 190 deixe que o computador escolha ao acaso a linha na qual o elefante deverá andar.

Acrescente mais três linhas que lhe darão controle sobre o elefante de modo que ele não esbarre nas árvores:

235 — O TK95 examina o teclado para saber qual a tecla que você está pressionando. Ele coloca o caractere correspondente à tecla em d\$.

242 — Se você pressionou "6", e o elefante não estiver na última linha da tela, o elefante desce.

244 — Se você pressionou "7" e o elefante não está exatamente na parte superior da tela, o elefante sobe.

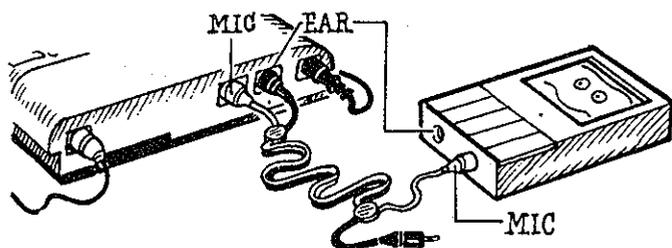
Volte ao programa "Areias Movediças" no Capítulo 1 ele poderá ser útil em relação a este assunto.

# 18

## Mais Tarefas para seu Gravador

Quando você carrega um programa em seu computador a partir de um gravador, é ótimo que o jogo comece imediatamente sem que você tenha que pressionar RUN. Para que isto aconteça você deverá gravar o programa de uma maneira especial.

Conecte o seu TK95 ao gravador assim:



Observe a listagem de seu programa na tela e decida em que linha o programa deve começar a ser rodado. Normalmente será a primeira linha, a linha 10. Para gravar "elefante" de modo que ele entre imediatamente ao ser carregado digite

SAVE "elefante" LINE 10

LINE ESTÁ EMBAIXO DA  
TECLA 3. USE O CURSOR E  
E SYMBOL SHIFT



Coloque o gravador na modalidade gravar e pressione qualquer das teclas no teclado do seu computador. Quando desaparecerem as listas coloridas e surgir a mensagem "OK", rebobine a fita e coloque o pino EAR em seu orifício de ligação. Verifique agora sua gravação digitando

VERIFY "elefante"

Pressione ENTER e ajuste o gravador para "play"



Quando você quiser usar o "elefante" carregue-o no TK95 da maneira usual como

LOAD "elefante"

ou

LOAD ""

Quando um programa está entrando seu nome surge na tela. Para ter certeza de que o nome não estará mais na tela quando o jogo começar, acrescente um comando CLS próximo ao início do programa.

NA LINHA 10 'ELEFANTE' TEM  
CLS  
ANTES QUE EU DESENHE AS  
ÁRVORES



Quando você estiver escrevendo um programa que você pretende gravar desta forma comece assim:

```
10 PAPER 7: INK: CLS
20 PRINT FLASH 1; AT 10,2; "PARE
A FITA PRESSIONE ENTER"
30 INPUT z$: CLS
```

Se você tiver projetado uma apresentação na tela particularmente interessante, talvez queira gravá-la sozinha, separada do resto do programa.

Aqui está um programa denominado "Arte Moderna"

```
10 REM Arte Moderna
20 INPUT "Cor para PAPER? ": p
30 IF p < 0 OR p > 7 THEN GO TO 20
40 PAPER p: CLS
50 FOR a=1 TO 50
60 LET i=INT (RND*7)
70 LET s=INT (RND*(100-a))+1
80 LET x=INT (RND*(256-s))
90 LET y=INT (RND*(176-s))
100 FOR q=0 TO s
110 PLOT INK i;x,y+q
120 DRAW INK i;s,0
130 NEXT q
140 NEXT a
```

SCREEN\$ É A MESMA PALAVRA  
QUE USAMOS NO CAPÍTULO 15  
MAS AQUI TEM OUTRO USO



Digite e rode-o algumas vezes. Quando chegar a uma representação na tela que você queira gravar digite

```
SAVE "arte" SCREEN$
```

É claro que você não precisa denominá-la "arte". Escolha qualquer nome que você considere adequado a sua representação. Na tela, uma vez que tanto as formas quanto as cores devem ser gravadas, o processo de armazenamento é demorado. Você não pode usar VERIFY para uma tela completa. Ao rodar a fita as palavras

Bytes: arte

aparecem na tela apagando uma parte do desenho. Por esta razão quando o TK95 compara aquilo que está na fita com o que está na tela verifica que são diferentes.



BYTES SÃO UNIDADES  
DE INFORMAÇÃO

Para que o trabalho de arte volte a passar da fita para a tela digite

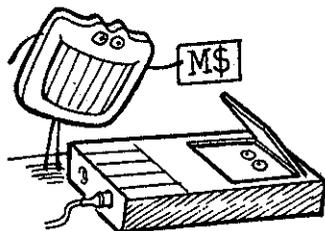
```
LOAD "arte" SCREEN$
```

Quando o computador encontrar o lugar certo na fita ele escreverá

Bytes: arte

e então você verá o desenho aparecer em preto e branco. A maneira pela qual ele vai aparecendo, em listas, é bastante surpreendente. Ao terminar o computador vai começar a colorir.

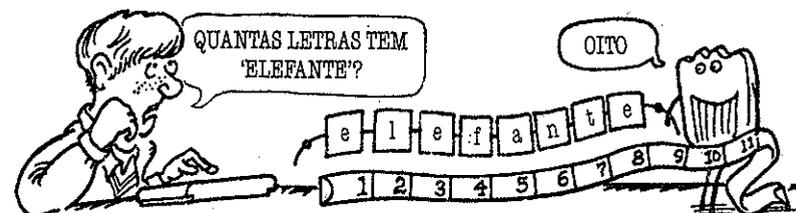
Você pode gravar não só um programa completo ou uma representação visual na tela mas também os dados de uma matriz. No Capítulo 21 falaremos mais sobre este assunto ao gravarmos uma mensagem em código.



# 19

## Medir e Cortar um String

O TK95 pode medir um string e dizer seu comprimento — em outras palavras, quantos caracteres ele tem.



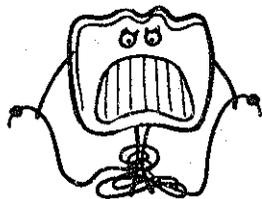
A palavra em BASIC para comprimento (length) é LEN, que você encontrará em verde em cima da tecla K. Digite

```
PRINT LEN "computar"
```

e observe qual o número que aparece na tela.

Aqui está um programa para medir palavras. Você digita qualquer palavra e o computador conta as letras.

```
10 INPUT "Quero medir uma palavra";w$
20 PRINT w$
30 PRINT "Quantos caracteres tem os aqui"
40 PRINT "O string é";LEN w$
```



Se você quiser que o computador sublinhe uma mensagem na tela é preciso que ele saiba o comprimento dela, neste caso LEN é muito útil. Em "Palavras luminosas", o TK95 sublinha a mensagem com estrelas coloridas e intermitentes. Ele vai também centralizar as palavras na tela deixando espaços iguais antes e depois.

```
10 REM Palavras luminosas
20 PAPER 7: INK 0: CLS
30 INPUT "Por favor digite sua
mensagem";m$
40 IF LEN m$ > 32 THEN RUN
50 LET comprimento=LEN m$
60 LET coluna=(32-comprimento)/2
70 PRINT AT 10, coluna;m$
80 PRINT AT 11, coluna;
90 FOR s=1 TO comprimento
100 LET c=INT (RND*7)
110 PRINT FLASH 1; INK c;"*";
120 NEXT s
```



A LINHA 60 CALCULA  
ONDE COMEÇAR A ESCREVER

Além de medir um string o TK95 pode também cortá-lo. Ele pode eliminar um só caractere ou um bloco com vários caracteres. Você é quem escolhe.



Coloque uma palavra em w\$ e faremos com que o TK95 a corte de várias maneiras.

```
LET w$"ferramenta"
```

Você pode pedir a quinta letra, assim:

```
PRINT w$(5)
```

Para obter a primeira letra digite

```
PRINT w$(1)
```

O que você acha que

```
PRINT w$(LEN w$)
```

apresentaria?



w\$ TEM DEZ LETRAS  
PORTANTO LEN w\$ é 10

Com "ferramenta" ainda em w\$, vamos pedir ao computador para apresentar um grupo de letras do meio da palavra. Digite

```
PRINT w$(2 TO 5)
```

e o TK95 apresentará a segunda, terceira, quarta e quinta letras.



VOCÊ JÁ ENCONTROU TO ANTES. ESTÁ EM VERMELHO NA TECLA F

Se você quiser as cinco primeiras letras de w\$ poderá dizer ao computador para

```
PRINT w$(1 TO 5)
```

Esta é uma outra maneira de fazer isto

```
PRINT w$ (TO 5)
```



SE VOCÊ OMITIR O NÚMERO ANTES DE TO, EU VOU SUPOR QUE VOCÊ QUER QUE EU COMECE NO INÍCIO DA PALAVRA

Agora tente

```
PRINT w$ (6 TO)
```

Se omitirmos o número depois de TO, o computador continuará escrevendo até o final da palavra. Verifique o que acontece se você digitar

```
PRINT w$(TO)
```

*Para você fazer*

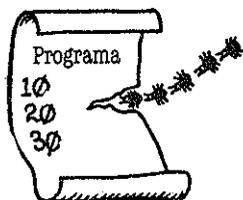
Faça um programa onde o computador pergunte seu nome e depois responde, "Muito prazer", e em seguida escreve o seu nome. Seja qual for o tamanho do nome ele não pode ser dividido em duas linhas.



"MUITO PRAZER" TEM 12 CARACTERES. SE O NOME TIVER MAIS DE 20 LETRAS PRECISO LEVÁ-LO PARA A PROXIMA LINHA ANTES DE ESCREVER

# 20 Bugs

Bugs são criaturinhas desagradáveis que se insinuam em seus programas e impedem que funcionem de modo correto.



É preciso considerar que eles nunca aparecem sem uma boa razão, e esta razão é muito simples, você cometeu erros ao escrever ou ao digitar o programa.

Muitas vezes o TK95 pode descobrir um destes errinhos e dizer a você onde está e como foi. Aqui está um programa com um bug na linha 20:

```
10 INPUT "Qual e o seu nome?"
;n$
20 PRINT "Alo ";m$
```

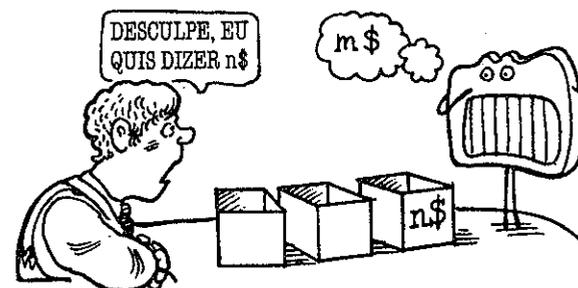
Execute e digite seu nome. Quando chegar na linha 20 o TK95 vai escrever

2 variable not found, 20:1

Isto se chama 'Mensagem de erro'. Você vai encontrar uma lista completa de mensagens de erro no manual do seu

TK95. O número 2 antes da mensagem é um código para ajudar você a encontrar a mensagem mais rapidamente.

Os números depois da mensagem indicam que o computador parou na linha 20, na primeira instrução da linha. O TK95 enviou a mensagem 'Variable not found', porque dissemos a ele que procurasse no bloco m\$ mas não tínhamos um bloco denominado m\$.



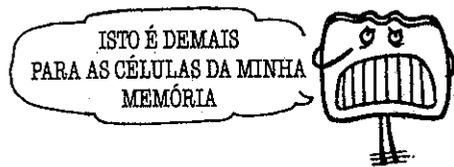
Os bugs ocorrem muitas vezes quando estamos lidando com uma matriz e se esquecemos quantos compartimentos ela tem. A palavra para compartimento no seu TK95 é 'subscript'. Isto você vai poder observar quando rodar este programa:

```
10 DIM a$(8)
20 LET a$="parabens"
30 PRINT a$(10)
```



Números podem provocar a entrada de erros no programa. Esta é uma das formas:

```
10 PRINT 99999999999999999999*
99999999999999999999
```



Um número menor também pode ocasionar o aparecimento de um bug se não for adequado ao local onde você o colocou. Tente isto:

```
10 PRINT AT 22,0;" Aqui estou"
```



Aqui estão mais dois programas contendo erros:

```
10 CIRCLE 150,80,100
```



e

```
10 PRINT CHR$ -1
```



Para você fazer:

Veja se pode encontrar os erros destes programas. Corrija os erros.

```
10 REM acrescentando programa
20 FOR a=1 TO 5
30 INPUT "Digite um numero.";nu
mero
40 LET total=total+numero
50 NEXT a
60 PRINT "O total e;total"
```



```
10 DIM f$(5,10)
20 FOR a=1 TO 6
30 READ f$(a)
40 NEXT a
50 DATA "margarida";"cravo";"ro
sa";"petunia";"violeta"

10 PLOT 100,80: DRAW 100,50
20 DRAW 0,-100: DRAW -100,-50
30 DRAW 0,100
```



# 21

## Codemaster

Com este programa você pode gravar uma mensagem em código para mandar para um amigo. Você poderá também decodificar a resposta.

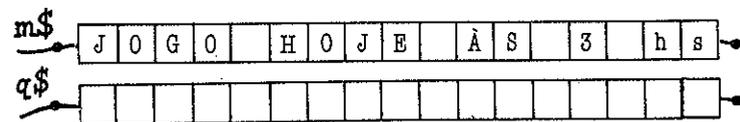


A primeira parte do programa é o 'menu', onde você escolhe se vai codificar ou decodificar uma mensagem.

```
10 REM Codemaster
20 CLS
30 PRINT AT 0,0; "Codemaster"
40 PRINT AT 2,0;"Digite"
50 PRINT AT 4,0;"1 para codificar
uma mensagem"
60 PRINT AT 6,0;"2 para decodifi
car uma mensagem"
70 PRINT AT 8,0;"3 para parar"
80 INPUT a$
90 CLS
100 IF a$="1" THEN GO SUB 140
```

```
110 IF a$="2" THEN GO SUB 350
120 IF a$="3" THEN STOP
130 GO TO 20
```

Em seguida vem a rotina 'codificar' que vai da linha 130 a linha 340. Sua mensagem vai para m\$ e o TK95 reserva uma matriz chamada q\$, com o mesmo comprimento que m\$, para a qual vai a mensagem codificada.



```
140 REM Codigo
150 INPUT "Digite sua mensagem
";m$
160 IF m$="" THEN RETURN
170 PRINT m$
180 DIM q$(LEN m$)
190 INPUT "Digite a palavra cha
ve";c$
200 IF c$="" THEN GO TO 190
```

Você vai precisar escolher uma 'palavra-chave' — qualquer uma — que você queira que irá para c\$. O computador vai então ampliar c\$ até que esteja do mesmo tamanho ou maior do que a mensagem.

```
210 IF LEN c$ >= LEN m$ THEN GO
TO 240
220 LET c$=c$+c$
230 GO TO 210
```



Agora codifique a mensagem:

```

240 FOR a=1 TO LEN m$
250 LET q$(a)=CHR$(CODE m$(a)+
CODE c$(a))
260 NEXT a
270 PRINT "A mensagem diz:—"
' q$

```



Por último o computador vai perguntar se você quer gravar sua mensagem em fita:

```

280 INPUT "Voce quer gravar em fi
ta? ";z$
290 IF z$="" THEN GO TO 280
300 IF z$(1) < >"y" THEN RETURN
310 INPUT BRIGHT 1;"Posicione o
gravador e terminais. Entao pres
sione ENTER. ";z$
320 SAVE "mensagem" DATA q$()
330 INPUT "Pare o gravador e pres
sione ENTER. ";z$
340 RETURN

```

A gravação é feita na linha 320. "mensagem" é o nome do arquivo e DATA q\$(0) diz ao computador para passar para a fita toda a informação a partir de q\$.



A sub-rotina 'decodificar' começa na linha 350. Ela é praticamente igual à de codificação mas em sentido inverso.

```

350 REM Decodificar
360 PRINT "Posicione o gravador
e terminais."
370 PRINT "Coloque o gravador
em play."
380 PRINT "Pressione ENTER."
390 INPUT z$
400 LOAD "mensagem" DATA q$()
410 CLS
420 PRINT "A mensagem codificada diz:—" ' q$
430 DIM m$(LEN q$)
440 INPUT "Pare o gravador e digi
te a palavra chave. ";c$
450 IF c$="" THEN GO TO 440
460 IF LEN c$ > = LEN q$ THEN GO
TO 490
470 LET c$=c$+c$
480 GO TO 460
490 FOR a=1 TO LEN q$
500 LET m$(a)=CHR$(CODE q$(a)-
CODE c$(a))
510 NEXT a

```

```

520 PRINT " " "A mensagem diz:—"
'm$
530 INPUT "Pressione ENTER. ";z$
540 RETURN

```

CERTIFIQUE-SE DE QUE  
NÃO ESTÁ EM CAPS LOCK  
QUANDO RODAR O PROGRAMA



APOSTO QUE NINGUÉM  
DESCOBRIRÁ  
MEU CÓDIGO SEM SABER A  
PALAVRA CHAVE

# 22

## Campo Minado

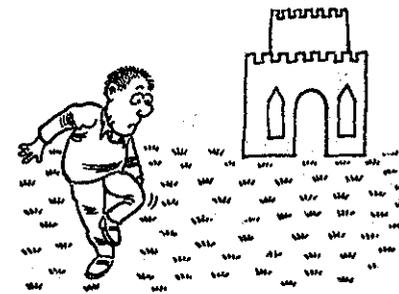
Há muitos séculos os habitantes de um castelo fortificado colocaram minas no pátio bem defronte à entrada. Ao longo dos anos estas minas vêm gradualmente chegando à superfície.

Ao realizar uma missão secreta naquela região você foi mordido por uma cobra venenosa. O único antídoto está no castelo, mas para chegar lá é preciso atravessar o pátio, onde as minas, uma por uma, estão chegando à superfície.

Pressione

- a para movimentar para cima
- z para movimentar para baixo
- n para a esquerda
- m para a direita

Será que você pode chegar ao castelo antes que seu tempo se esgote?



```

10 campo minado
20 LET Score maximo=0: LETn$=""
30 FOR c=144 TO 145

```



```

370 IF i$="m" AND coluna < 30 THE
N LET nova_coluna=coluna+1
380 PRINT AT linha,coluna;" "
390 LET linha=nova linha: LET co
luna=nova coluna
400 IF SCREEN$(linha,coluna)="*
" THEN GO SUB 420: GO SUB 630:
GO TO 110
410 GO TO 240

```



```

420 REM Morto
430 PRINT INK 3:AT,linha,coluna
;"B"
440 RESTORE 490
450 FOR t=1 TO 10
460 READ v,p
470 SOUND v,p
480 NEXT t
490 DATA .2,15,.2,14,.4,14,.2,1
5,.2,14,.4,14
500 DATA .2,15,.2,14,.4,14,.4,1
0
510 RETURN

520 REM Sucesso
530 RESTORE 580
540 FOR t=1 TO 10
550 READ v,p
560 SOUND v,p

```

```

570 NEXT t
580 DATA .2,5,.2,7,.4,9,.4,5,.4
,5
590 DATA .2,7,.2,9,.4,10,.4,14,
.4,14
600 RETURN

610 REM Recordista
620 IF tempo > escore maximo THEN
GO SUB 700
630 CLS
640 PRINT INK 2;AT 2,2;"MINEFI
ELD"
650 PRINT INK 0;AT 5,2;"STOP SC
CORER:", escore maximo
660 PRINT INK 0;AT 7,2;n$
670 PRINT INK 2;AT 10,2;"PRES
SIONE QUALQUER TECLA PARA JOGAR
OUTRA VEZ"
680 PAUSE 0
690 RETURN

700 REM Novo recordista
710 CLS
720 LET escore maximo=tempo
730 PRINT INK 2;AT 2,2;"Voce e
o novo recordista."
740 PRINT INK 0;AT 4,2;"Por favor digite seu nome."
750 INPUT n$
760 RETURN

```

VOCÊ PRECISA DE MIM  
PARA EXECUTAR O PROGRAMA



# 23

## Parceiros



As agências que promovem encontros totalmente automatizadas existem há muitos anos e prestam bons serviços encontrando parceiros adequados para seus clientes. Nosso programa naturalmente, é muito simples se o compararmos com os que estas agências utilizam mas assim mesmo ele poderá dar a você uma idéia de como o computador escolhe os pares.

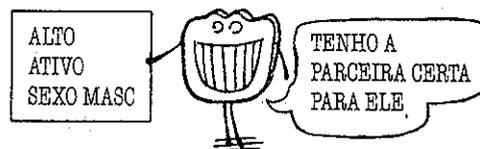
Os dados sobre os parceiros estão no final do programa linhas 680 a 770. Possuímos cinco informações a respeito deles:

Nome,  
Sexo — masc/fem,  
Cor do cabelo — escuro/louro/ruivo,  
Estatura — baixa/média/alta,  
Temperamento — ativo/sossegado,

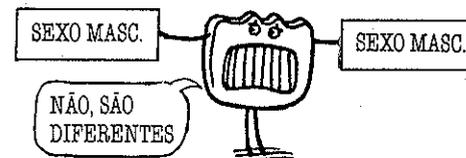
A matriz que contém esta informação é d\$ e é estabelecida na linha 20. Tem espaço para dez registros, cada um com cinco campos. A maior palavra que pode ser armazenada possui oito letras.

| d\$ | 1<br>Nome | 2<br>Sexo | 3<br>Cabelos | 4<br>Estatura | 5<br>Temperamento |
|-----|-----------|-----------|--------------|---------------|-------------------|
| 1   | SIMON     | MASC      | ESCUROS      | BAIXA         | ATIVO             |
| 2   | GARY      | MASC      | LOUROS       | MÉDIA         | ATIVO             |
| 3   | AMANDA    | FEM       | ESCUROS      | MÉDIA         | SOSSEGADO         |
| 4   | VITORIA   | FEM       | ESCUROS      | ALTA          | ATIVO             |
| 5   | ANDREW    | MASC      | ESCUROS      |               |                   |
| 6   | RICHARD   | MASC      |              |               |                   |
| 7   | PAUL      | MASC      |              |               |                   |

O computador faz perguntas a seu respeito e compara suas respostas com a informação disponível a respeito dos parceiros.



Aqui está o programa. Use o cursor L e não o C quando rodá-lo. Os dados que o computador possui estão em letras minúsculas e seu input deve estar assim também para que o TK95 possa ser capaz de fazer uma comparação correta.



10REM Parceiros

20 DIM d\$(10,5,8)

30 DIM g\$(8): DIM t\$(8) ; DIM h

```

$(8)
40 LET m=0
50 REM Leia DATA
60 FOR r=1 TO 10
70 FOR f=1 TO 5
80 READ d$(r,f)
90 NEXT f
100 NEXT r

110 REM Introducoes
120 BORDER 6: PAPER 7: INK 1: C
LS
130 PRINT AT 2,2;"Bem-vindo aos
encontros programados!"
140 PRINT AT 5,2;"Por favor apre
sente-se"
150 PRINT AT 7,2;"digitando seu
nome."
160 INPUT n$
170 CLS
180 PRINT AT 2,2;"Obrigado ";n$
190 PRINT AT 5,2;"Voce e homem
ou mulher?"
200 INPUT s$
210 CLS
220 IF s$ < > "homem" AND s$ < > "mu
lher": THEN GO TO 190
230 IF s$="homem" THEN LET g$="
mulher": LET p$="Ela"
240 IF s$="mulher" THEN LET g$
="homem": LET p$="Ele"
250 PRINT AT 2,2;"Agora preciso
saber alguma coisa"
260 PRINT AT 4,2;"sobre voce."
270 PRINT AT 7,2;"Voce e"

```

```

280 PRINT INK 2;AT 9,2;"sossega
do"; INK 0;" ou "; INK 2;"ati
vo"; INK 0;"?"
290 INPUT v$
300 CLS
310 IF v$ < > "sossegado" AND v$ < > "ativo" THEN
GO TO 270
320 LET t$=v$
330 PRINT AT 2,2;"Voce e"
340 PRINT INK 2;AT 5,2;"alto"
350 PRINT INK 2;AT 7,2;"medio"
360 PRINT INK 0;AT 9,2;"ou"
370 PRINT INK 2;AT 11,2;"baixa
","?"
380 INPUT j$
390 CLS
400 IF j$ < > "alto" AND j$ < > "medi
a" AND j$ < > "baixa" THEN GO TO
330
410 LET h$=j$
420 BORDER 2
430 PRINT AT 2,2;"Obrigado, ";
n$
440 PRINT AT 4,2;"Agora vou pro
curar em meus"
450 PRINT AT 6,2;"registros a pes
soa adequada."
460 INPUT "Por favor pressione
ENTER";z$
470 CLS

480 REM Verificar dados
490 FOR p=1 TO 10
500 IF d$(p,2)=g$ AND d$(p,5)=t
$ AND d$(p,4)=h$ THEN GO SUB 58

```

```

0
510 NEXT p
520 IF m=1 THEN BORDER 4: PRINT AT 2,2;"Que tal um encontro?":
STOP
530 BORDER 0

540 REM Nenhum parceiro
550 PRINT AT 5,2;"Desculpe, "n$
560 PRINT AT 7,2;"Nao achei ne
nhum parceiro para voce."
570 STOP

580 REM Um parceiro
590 LET m=1
600 BORDER 3
610 PRINT AT 2,2;d$(p,1);" e uma
possibilidade."
620 PRINT AT 5,2;p$;"e";d$(p
,4);"em estatura."
630 PRINT AT 7,2;p$;"e";d$(p
,5)
640 PRINT AT 9,2;"e tem ";d$(
p,3);" cabelos."
650 INPUT "Por favor pressione ENTER.
";z$
660 CLS : RETURN

670 REM Dados
680 DATA "Simon","homem","moreno
","baixo","ativo"
690 DATA "Garry","homem","louro
","media","ativo"
700 DATA "Amanda","mulher","mo
rena","media","sossegada"

```

```

710 DATA "Victoria","mulher","mo
rena","alta","ativa"
720 DATA "Andrew","homem","more
no","alto","ativo"
730 DATA "Richard","homem","mo
reno","medio","ativo"
740 DATA "Paul","homem","ruivo"
,"alto","sossegado"
750 "Jane","mulher","more
na","baixa","ativa"
760 DATA "Lisa","mulher","loura
","media","ativa"
770 DATA "Diana","mulher","mo
rena","sossegada"

```



TALVEZ VOCÊ QUEIRA DESCREVER SEUS AMIGOS — linha 680 A 770. SE ALGUM DELES TIVER UM NOME COM MAIS DE OITO LETRAS TROQUE OS 8 NAS LINHAS 20 E 30 PARA UM NÚMERO MAIOR.

BARTHOLOMEU

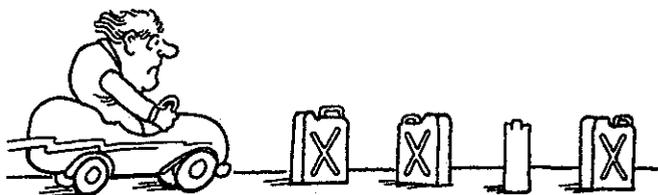


VOU TROCAR OS 8 PARA 11

# 24

## Circuito

Vinte latas de óleo foram deixadas na pista do autódromo. Você vai dar uma volta completa na pista, sem esbarrar nas latas e sem bater nas barreiras laterais.



Ao começar, seu carro está andando para a direita. Para mudar de direção pressione

n para a esquerda  
m para a direita  
a para cima  
z para baixo



SE VOCÊ QUISER USAR TECLAS  
DIFERENTES  
ALTERE AS LETRAS NAS ASPAS  
NAS  
LINHAS 170 A 200

O B na linha 50 precisa do cursor G, o mesmo acontece com o A na linha 230, com o B da linha 240 e com o C da linha 960.

```
10 REM Circuito
20 GO SUB 1000: REM Caractere Gr
afico
30 GO SUB 800: REM Pista
40 REM Variaveis
50 LET c$="B"
60 LET linha=19: LET coluna=15
70 LET movelinha=0: LET moveco
luna=1
80 LET d$="m"
90 LET score=0

100 REM Corrida
110 PRINT INK 2; AT linha, colu
na; c$
120 SOUND .01, -12
130 LET score=score+1
140 PRINT INK 0; AT 10, 12; "ESCO
RE "; score
150 LET i$=INKEY$
160 IF i$ < > " " THEN LET d$=i$
170 IF d$="n" THEN LET movecol
una=-1; LET movelinha=0
180 IF d$="m" THEN LET movecol
una=-1: LET movelinha=0
190 IF d$="a" THEN LET moveli
nha=-1: LET movecoluna=0
200 IF d$="z" THEN LET moveli
nha=1: LET movecoluna=0
210 PRINT AT linha, coluna; " "
220 LET linha=linha+movelinha:
LET coluna=coluna+movecoluna
230 IF coluna < =3 OR coluna > = 28
THEN LET c$="A"
240 IF linha < = 3 OR linha > = 18
```

```

THEN LET c$="B"
250 IF ATTR (linha,coluna)=32 OR
ATTR (linha,coluna)=57 THEN GO
TO 700: REM colisao
260 GO TO 110

```

ATTR SERVE PARA VERIFICAR SE  
VOCÊ DERRUBOU UMA LATA OU  
SE  
SAIU DA PISTA



```

700 REM Colisao
710 PRINT INK 2; FLASH 1; AT li
nha,coluna;c$
720 FOR p=12 TO -12 STEP -1
730 SOUND .01,p
740 NEXT p
750 INPUT "Pressione ENTER para
uma nova corrida. ";z$
760 RUN

```

```

800 REM Tela
810 BORDER 5: PAPER 4: CLS
820 PAPER 7: INK 0
830 FOR r=1 TO 3
840 PRINT AT r,1;"
"
850 NEXT r
860 FOR r=4 TO 17
870 PRINT AT r,1;" ";AT r,28;
" "

```

```

880 NEXT r
890 FOR r=18 TO 20
900 PRINT AT r,1;"
"
910 NEXT r

```



NAS LINHAS 840 E 900 EXISTEM  
30 ESPAÇOS ENTRE AS ASPAS. NA  
LINHA 870 EXISTEM TRÊS

```

920 REM Perigos
930 FOR h=1 TO 20
940 LET hr=INT (RND*19)+1
950 LET hc=INT (RND*29)+1
960 PRINT INK 1; AT hr,hc;"C"
970 NEXT h
980 RETURN

```

```

1000 REM Caracteres Graficos
1010 RESTORE 1090
1020 FOR c=144 TO 146
1030 FOR a=0 TO 7
1040 READ n
1050 POKE USR CHR$ c+a,n
1060 NEXT a
1070 NEXT c
1080 RETURN
1090 REM Carro para cima/para
baixo
1100 DATA 186,254,186,56,56,186,
254,186

```

```
1110 REM Carro para a esquerda/
para direita
1120 DATA 0,231,66,255,255,255,6
6,231
1130 REM lata de óleo
1140 DATA 0,56,40,60,60,60,60,0
```

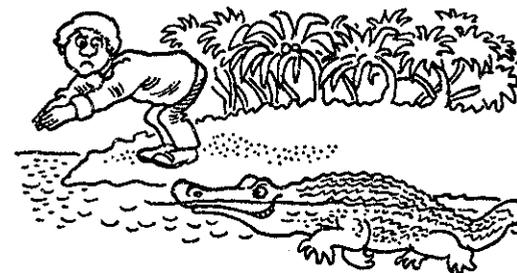


PARA UM NOVO TIPO DE  
PERIGO, MUDE OS NÚMEROS  
DA LINHA 1140

# 25

## Aventura na Selva

Este programa é um exemplo de uma aventura do tipo texto. Isto significa que ao jogar, você toma parte numa estória escrita na tela. Você vai encontrar quatro perigos enquanto estiver atravessando a selva a caminho de seu acampamento. Se você vai escapar ou não, depende em parte de estar levando o equipamento adequado, e em parte de ter sorte.



```
10 REM Aventura na Selva
20 BORDER 5; PAPER 7: INK 0: C
LS
30 GO SUB 890: REM Descricao
40 GO SUB 320: REM variaveis
50 INPUT BRIGHT 1;"Pressione ENTE
R para continuar.";z$
60 CLS

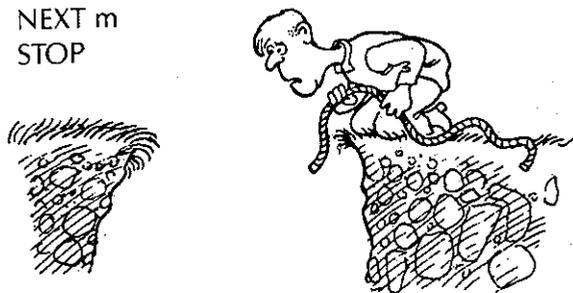
70 FOR x=1 TO 4
80 IF y<2 THEN GO SUB 410: RE
M Ofereca equipamento
90 REM Escolha perigo
```

```

100 LET h=INT (RND*4)+1
110 IF g(h)=1 THEN GO TO 100
120 LET g(h)=1
130 BORDER h: INK h: CLS
140 PRINT "A sua frente esta"
h$(h)
150 PRINT
160 LET v$="n"
170 IF d(h)=1 THEN GO SUB 590:
REM Usar equipamento?
180 IF d(h)=0 OR v$="n" THEN G
O SUB 710: REM Arrisque
190 IF v$="y" OR v$="Y" THEN L
ET d(h)=0: LET y=y-1
200 CLS
210 NEXT x

220 REM Parabens
230 BORDER 5: INK 2: CLS
240 PRINT "PARABENS!"
250 PRINT "Voce chegou ao seu
campo" "a salvo."
260 IF v=0 THEN STOP
270 PRINT "Voce ainda tem"
280 FOR m=1 TO 4
290 IF d(m)=1 THEN PRINT i$(m)
300 NEXT m
310 STOP

```



```

320 REM Variaveis
330 DIM h$(4,32): DIM i$(4,32)
340 DIM a$(4,64): DIM w$(4,64)
350 DIM f$(4,32): DIM u$(4,32)
360 DIM g(4): DIM d(4): LET y=0
370 FOR z=1 TO 4
380 READ h$(z),i$(z),a$(z),w$(z
),f$(z),u$(z)
390 NEXT z
400 RETURN

```



```

410 REM Oferecer equipamento
420 LET i=INT (RND*4)+1
430 IF d(i)=1 THEN RETURN
440 PRINT "A sua frente esta
" i$(i)" BRIGHT 1;"Voce quer
pegar?"
450 DIM z$(1): INPUT z$
460 PRINT
470 IF z$="y" OR z$="Y" THEN G
O SUB 520
480 IF z$ < > "y" AND z$ < > "Y" THEN
PRINT "Voce esta partindo" i$(i)
490 INPUT BRIGHT 1; Pressione E
NTER ";z$

```

```

500 CLS
510 RETURN

520 REM Pegue o equipamento
530 LET y=y-1: LET d(i)=1
540 PRINT "Voce agora esta levando"
550 FOR m=1 TO 4
560 IF d(m)=1 THEN PRINT i$(m)
570 NEXT m
580 RETURN
590 REM Usar equipamento?
600 "Felizmente voce tem" i$
(h)' BRIGHT 1;"Voce quer usa-lo"
610 DIM v$(1): INPUT v$
620 IF v$="y" OR v$="Y" THEN GO
SUB 0640
630 RETURN

640 REM Usar equipamento
650 PRINT
660 LET r=INT (RND*6)
670 IF r < >0 THEN PRINT f$(h)
680 IF r=0 THEN GO TO 820
690 INPUT "Pressione ENTER ";z$
700 RETURN
710 REM Arrisque
720 PRINT "Infelizmente, voce nao tem" i$(h)
730 PRINT
740 PRINT w$(h)
750 INPUT BRIGHT 1;"Pressione ENTER ";z$
760 PRINT
770 LET r=INT (RND*2)

```



```

780 IF r=0 THEN PRINT f$(h)
790 IF r < >0 THEN GO TO 820
800 INPUT BRIGHT 1;"Pressione ENTER ";z$
810 RETURN

820 REM FINAL
830 CLS
840 BORDER 0: INK 0
850 PRINT "Que azar!"
860 PRINT AT 2,0;u$(h)
870 PRINT AT 4,0;"Voce viajou "
;(x-1)/4*100;"%" "do caminho a
traves da selva"
880 STOP

890 REM Descricao
900 PRINT "Aventura na Selva"
910 PRINT 'TAB 5; Voce tem que
atravessar uma grande "flores
ta para chegar ao seu acampamen
to"
920 PRINT "Voce tera a oportun

```

idade'' ''de pegar certos itens n  
o''

940 PRINT "caminho, mas lembre-  
se'' ''so dois objetos de cada'' '' ve  
z."

950 PRINT ' TAB 5;"Quando for f  
eita uma'' '' pergunta, responda si  
m ou nao.'' '' Para uma acao mais  
rapida a'' '' qualquer momento, pr  
essione qualquer tecla."

960 RETURN

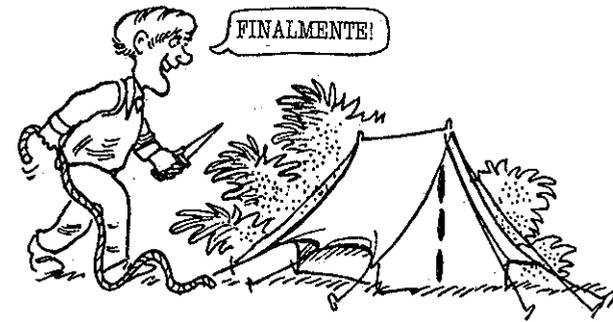
970 REM DATA

980 DATA "ravina profunda.";"c  
orda.";"Voce joga a corda e co  
meça a atravessar a ravina.";"Vo  
ce tenta pular.";"Voce chega sa  
o e salvo ao outro lado.";"Voc  
e escorrega, e cai no abismo."

990 DATA "um rio com crocodilos."  
;"uma balsa de borracha.";"Voce ench  
e a balsa e parte.";"Voce começa a nadar."  
;"Voce chega a outra margem.";"Voc  
e acaba na barriga do crocodilo."

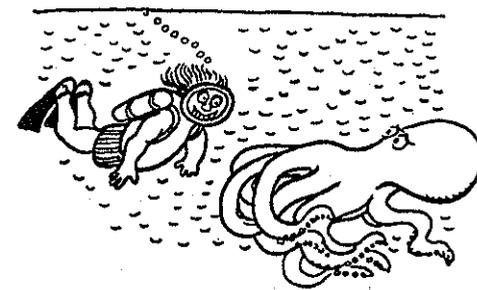
1000 DATA "uma terrivel cobra ve  
nenosa.";"um punhal.";"Voce ataca a cobra com a  
faca.";"Voce tenta estrangula-la.";"Presas envenenada  
s liquidam voce.";"A cobra esta  
morta a seus pes."

1010 DATA "um nativo enorme e fe  
roz.";"uma sacola com moedas de o  
uro.";"Voce oferece a sacola.";"Vo  
ce tenta fugir dele.";"Ele deixa  
voce passar.";"Voce serve de almo  
co."



Voce pode alterar algumas partes deste programa para tornar a aventura mais emocionante — uma aventura submarina, quem sabe? Altere a descrição nas linhas 910 e 920 e os dados nas linhas 980 a 1010. Cada linha de dados contém as palavras que podem ser necessárias para os perigos. Cada uma delas contém seis itens.

1. As coisas terríveis que você vai enfrentar.
2. O equipamento que pode auxiliar.
3. O que você faz se tiver o equipamento.
4. O que você faz se não tiver o equipamento.
5. O que acontece se você for bem-sucedido.
6. O que acontece se você fracassar.



# 26

## Cursosores e Teclas – Um Lembrete

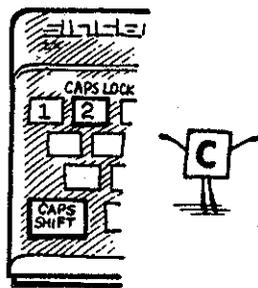
### Palavras-chave

As palavras que você vê em branco, sobre as teclas, são denominadas 'palavras-chave' (keywords). Quando você liga o seu TK95 e pressiona ENTER vai ver um K intermitente no canto esquerdo de sua tela. Com este K na tela, ao pressionar qualquer das teclas das três últimas carreiras do teclado você fará surgir na tela uma palavra-chave.



### Letras

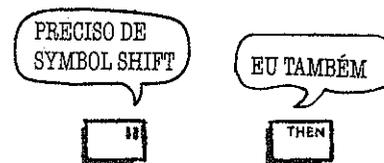
Com o cursor L na tela, ao pressionar uma tecla com uma letra você terá a letra minúscula. Para uma letra maiúscula, mantenha CAPS SHIFT pressionado enquanto digita a letra. Se você quiser escrever todo seu texto em letras maiúsculas, passe para a modalidade CAPS LOCK: mantenha CAPS SHIFT pressionada enquanto aperta CAPS LOCK.



O cursor mudará para C. Para voltar ao cursor L, e letras minúsculas, pressione CAPS SHIFT e outra vez CAPS LOCK.

*Símbolos em vermelho e palavras na parte superior das teclas.*

Para estes, você deve manter pressionado SYMBOL SHIFT em vermelho enquanto aperta a tecla.



*As palavras em verde na parte superior das teclas.*

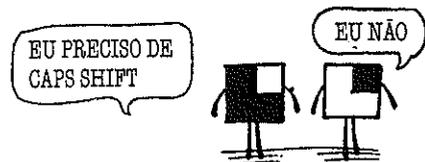
Para estas palavras use o cursor E. Pressione CAPS SHIFT e SYMBOL SHIFT juntos. Solte as duas e pressione a tecla abaixo da palavra que você deseja.

*As palavras em vermelho abaixo das teclas.*

Mude para o cursor E. Mantenha SYMBOL SHIFT pressionado e aperte a tecla imediatamente acima da palavra que você quer.

*Os caracteres gráficos para blocos.*

Estes são as formas quadradas nas teclas de 1 a 8. Para eles você vai precisar do cursor G. Mantenha CAPS SHIFT pressionada e aperte a tecla 9 — aquela onde GRAPHICS está escrito em cima. Solte CAPS SHIFT e então aperte a tecla 1. Você verá na tela um pequeno quadrado preto — a parte cinza no caractere gráfico na tecla 1. Mantenha CAPS SHIFT pressionado e aperte a tecla 1 outra vez. Desta vez vai ver um bloco em forma de L — a parte do caractere gráfico que está em branco na tecla.



Para voltar ao cursor normal, aperte 9 outra vez.

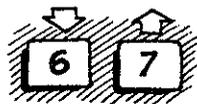
### *Alterando uma linha de programa*

Quando você digita uma linha de programa vê um símbolo como este > entre o número da linha e a primeira palavra da linha. Isto se chama "cursor do programa". Antes de poder realizar a alteração de uma linha você precisa movimentar este cursor para a linha em questão.

Existem duas formas de fazer isto. Uma é digitar LIST e então o número da linha que você quer alterar.



A outra é usar as setas.



Mantenha CAPS SHIFT pressionada e aperte a tecla 6. O cursor do programa descerá. Para que ele suba, use CAPS SHIFT com a tecla 7.

Com o cursor do programa na linha certa mantenha CAPS SHIFT pressionada e aperte EDIT (tecla 1). A linha descerá para a parte inferior da tela. Use a seta da esquerda ou da direita (teclas 5 e 8), com CAPS SHIFT, para levar o cursor K ou o L para a parte da linha que você quer alterar. Digite então algumas letras adicionais, ou use CAPS SHIFT e DELETE (tecla 0) para eliminar.

### *Inverse e true vídeo*

Se você pressionar CAPS SHIFT e INV. VIDEO, isto diz ao TK95 que inverta as cores de INK e de PAPER. Em vez de letras pretas em fundo branco, você terá letras brancas em fundo preto. CAPS SHIFT e TRUE VIDEO farão com que tudo volte ao normal.

### *O cursor E e a carreira de cima*

Pressionando uma tecla da carreira de cima quando o cursor E está na tela você poderá fazer coisas estranhas com sua listagem — alterar sua cor ou torná-la intermitente por exemplo.

Isto acontece muitas vezes por engano. Se acontecer digite DELETE algumas vezes e tudo voltará ao normal. Você encontrará uma tabela no seu manual, que dará maiores detalhes a esse respeito.

# 27

## Sugestões

Em alguns dos capítulos anteriores existe uma seção denominada *Para você fazer*. Nela existem perguntas e idéias a respeito de programas. Existem muitas maneiras de solucionar estes quebra-cabeças, mas em todo caso aqui estão algumas sugestões.

### Capítulo 1

#### Variáveis numéricas

score mantém um registro dos pontos marcados.  
yc é o número da coluna em que você está.  
pc a coluna da pista de aterrissagem.  
a um número entre 1 e 4 escolhido aleatoriamente pelo computador. Ele vai então usar este número para decidir se a pista de aterrissagem vai se mover para a esquerda ou para a direita, ou ainda permanecer no mesmo lugar.

#### Variáveis de string

c\$ sua nave.  
d\$ o caractere na tecla que você pressiona para movimentar sua nave.

### Capítulo 3

Aqui estão duas sugestões para o "cartão de aniversário" — uma versão simplificada.

```
10 REM Happy Birthday
20 REM estrelas
30 INK 2
40 PRINT AT 6,0;"*****"
*****"
50 PRINT AT 16,0;"*****"
*****"
60 REM palavras
70 INK 1
80 PRINT AT 8,2; Happy birthda
y to you"
100 PRINT AT 12,2;"Happy birthd
ay dear Thomas"
110 PRINT AT 14,2;"Happy birthd
ay to you"
120 REM melodia
130 FOR n=1 TO 25
140 READ d,p
150 SOUND d,p
160 NEXT n
170 REM data
180 DATA .5,0,.5,0,1,2,1,0,1,5,
2,4
190 DATA .5,0,.5,0,1,2,1,0,1,7,
2,5
200 DATA .5,0,.5,0,1,12,1,9,1,5
,1,4,1,2
210 DATA .5,10,.5,10,1,9,1,5,1,
7,2,5
```

e um mais complicado:

```
10 REM Happy Birthday
20 REM estrelas
30 FOR c=0 TO 31
```

```

40 LET i=INT (RND*7)
50 INK i
60 PRINT AT 0,c;"";AT 21,c;""
"
70 NEXT c
80 FOR r=1 TO 20
90 LET i=INT (RND*7)
100 INK i
110 PRINT AT r,0;"";AT r,31;""
"
120 NEXT r
130 REM imprimir cartao e tocar
melodia
140 FOR a=1 TO 4
150 READ a$,n
160 INK a: PRINT AT 7+a,2;a$
170 FOR b=1 TO n
180 READ d,p
190 SOUND d,p
200 NEXT b
210 PRINT
220 NEXT a
230 REM os dados
240 DATA "Happy birthday to you
",6
250 DATA .5,0,.5,0,1,2,1,0,1,5,
2,4
260 DATA "Happy birthday to you
",6
270 DATA .5,0,.5,0,1,2,1,0,1,7,
2,5
280 DATA "Happy birthday dear T
hommas",7
290 DATA .5,0,.5,0,1,12,1,9,1,5
,1,4,1,2

```

```

300 DATA "Happy Birthday to you
",6
310 DATA .5,10,.5,10,1,9,1,5,1,
7,2,5

```

#### Capítulo 4

```

10 REM idades
20 DIM a(5)
30 FOR p=1 TO 5
40 READ a(p)
50 NEXT p
60 DATA 42,39,18,15,11
70 PRINT "Por favor digite sua i
dade."
80 INPUT eu
90 LET n=0
100 FOR g=1 TO 5
110 IF a(g) > eu THEN LET n=n+1
120 NEXT g
130 PRINT
140 PRINT "O numero de pessoas"
150 PRINT "mais velhas do que v
oce e ":n

```

#### Capítulo 5

```

10 REM conta letras
20 DIM w$(20)
30 INPUT "Quero uma palavra.":w$
40 LET n=0
50 FOR c=1 TO 11
60 IF w$(c)="s" THEN LET n=n+
1

```

```

70 NEXT c
80 PRINT "EM ";w$
90 PRINT "A letra s aparece"
100 PRINT n:" vezes"

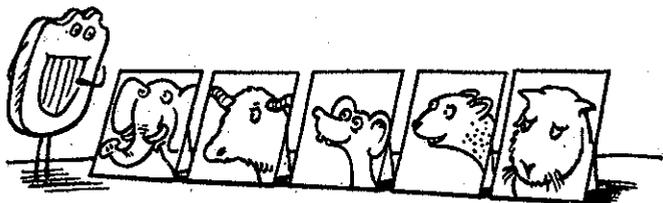
```

### Capítulo 6

```

10 REM animais
20 DIM $(5,10)
30 PRINT "Por favor digite o n 
me de ""cinco animais."" ""Nao ma
is de dez letras em cada."
40 PRINT
50 FOR a=1 TO 5
60 INPUT a$(a)
70 PRINT a$(a)
80 NEXT a
90 PRINT
100 LET b=INT (RND*5)+1
110 PRINT "O melhor e o":a$(b)

```



### Capítulo 7

```

10 DIM A$(10,3,20)
30 READ A$(R,1),A$(R,2),A$(R,3)
150 PRINT AT 10,0;"Quais os end
recos que voce quer?"
160 DIM N$(20)

```

```

200 IF A$(R,1)=N$ THEN PRINT A
T 6,0;A$(R,1);AT 8,0;A$(R,2);AT
10,0;A$(R,3)

```

Altere as linhas de dados (50 a 140) de modo que contemham o nome, endereco e telefone assim:

```

50 DATA "ANNE SMITH";"3 NEW ROAD";"709723"

```

### Capítulo 8

```

300 REM Direita
310 LET escore=escore+1
320 BORDER 2
330 PRINT AT 10,2;"Direita"
340 RESTORE 390
350 FOR m=1 TO 13
360 READ d,p
370 SOUND d,p
380 NEXT m
390 DATA .2,9,.2,9,.4,9,.2,6,.2
,6,.4,6
400 DATA .2,11,.1,11,.2,11,.2,9
,.2,7,.2,6,.4,4
410 RETURN
500 REM Errado
510 BORDER 0
520 PRINT AT 10,2;"Nao,e'";a$(a,
,2)
530 RESTORE 580
540 FOR m=1 TO 14
550 READ d,p
560 SOUND d,p
570 NEXT m

```

```

580 DATA 1,4,.5,5,.5,7,.5,7,.5,
5,.5,4,.5,2
590 DATA .5,0,.5,0,.5,2,.5,4,.7,
5,4,.25,2,1,2
600 RETURN

```

### Capítulo 9

```

10 REM Estrela de David
20 PLOT 90,60
30 DRAW 80,0
40 DRAW -40,70
50 DRAW -40,-70
60 PLOT 90,105
70 DRAW 80,0
80 DRAW -40,-70
90 DRAW -40,70

```



```

10 REM Rosto
20 PLOT 90,10
30 DRAW 0,10
40 DRAW -30,0
50 DRAW 0,100
60 DRAW 100,0
70 DRAW 0,-100
80 DRAW -30,0
90 DRAW 0,-10
100 PLOT 80,50
110 DRAW 0,-10
120 DRAW 60,0
130 DRAW 0,10
140 PLOT 80,100
150 DRAW 20,0
160 DRAW 0,-20

```



```

170 DRAW -20,0
180 DRAW 0,20
190 FOR y=80 TO 90
200 PLOT 90,y
210 DRAW 10,0
220 NEXT y
230 PLOT 120,100
240 DRAW 20,0
250 DRAW 0,-20
260 DRAW -20,0
270 DRAW 0,20
280 FOR y=80 TO 90
290 PLOT 130,y
300 DRAW 10,0
310 NEXT y

```

```

10 REM Passos
20 FOR a=10 TO 150 STEP 20
30 PLOT a,a
40 DRAW 20,0
50 DRAW 0,20
60 DRAW -20,0
70 DRAW 0,-20
80 NEXT a

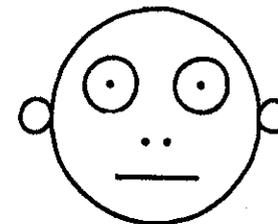
```

### Capítulo 10

```

10 REM Rosto Redondo
20 CIRCLE 120,80,50

```



```

30 PLOT 90,90
40 CIRCLE 90,90,10
50 PLOT 150,90
60 CIRCLE 150,90,10
70 PLOT 110,70
80 PLOT 130,70
90 PLOT 100,50
100 DRAW 40,0
110 CIRCLE 65,80,5
120 CIRCLE 175,80,5
200 REM Círculos
210 FOR a=1 TO 20
220 LET c=INT (RND*7)
230 LET x=INT (RND*150)+50
240 LET y=INT (RND*75)+50
250 LET r=INT (RND*40)+10
260 BORDER c: INK c: CIRCLE x,y
,r
270 SOUND .1,r
275 NEXT a
280 INPUT "Pressione ENTER",b$
290 RETURN

```

### Capítulo 11

A linha adicional é

```

150 IF POINT (x,y)=1 THEN GOTO
170

```

### Capítulo 13

A mensagem diz:

RUN OVER THE BORDER AND THEN  
GO TO THE NEXT CHECK POINT



### Capítulo 14

```

70 DATA 16,56,124,16,56,124,16
,16
80 DATA 60,60,24,255,24,60,102
,195
90 DATA 124,84,254,124,40,40,1
08,0
100 DATA 0,0,248,228,231,255,10
2,102

```

### Capítulo 15

```

10 REM Carneiro
20 FOR a=0 TO 7
30 READ n
40 POKE USR "a"+a,n
50 NEXT a
60 DATA 0,6,127,255,254,254,68
,68
70 REM Cerca
80 PAPER 4: INK 0: CLS

```



```

90 FOR f=0 TO 21
100 LET r=INT (RND*3)
110 IF r=0 THEN PRINT AT f,20;
" "
120 IF r< >0 THEN PRINT AT f,20
;" # "
130 NEXT f
140 REM Andar
150 INK 7
160 FOR c=0 TO 30
170 PRINT AT 10,c;"A"
180 PAUSE 10
190 IF SCREEN (10,c+1)=" # " THE
N STOP
200 PRINT AT 10,c;" "
210 NEXT c

```

### Capítulo 16

A linha adicional é

```
5 BRIGHT 1
```

Mude a linha 60 para

```
60 PRINT FLASH 1; INK 0; OVER
1;AT 10,c;"+"

```

### Capítulo 17

As duas linhas que devem ser alteradas são

```
120 BORDER 4: PAPER 7: CLS
220 IF ATTR (e,c+1)=60 THEN SO
UND 2,-10: STOP

```

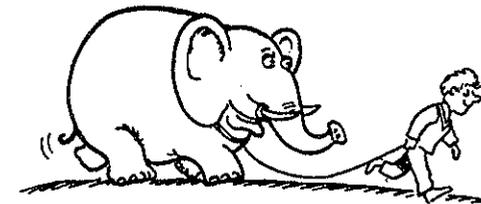
Esta é a linha que escolhe onde o elefante começa:

```
190 LET e=INT (RND*22)
```

Estas são as linhas que permitem que você controle o elefante:

```
235 LET d$=INKEY$
242 IF d$='6' AND e < 21 THEN LET
e=e+1
244 IF d$='7' AND e > 0 THEN LET
e=e-1

```



### Capítulo 19

```
10 INPUT "Por favor digite seu
nome.":n$
20 PRINT "Muito prazer";
30 IF LEN n$ > 7 THEN PRINT
40 PRINT n$

```

### Capítulo 20

Aqui estão as alterações que expulsarão os bugs:

*Acrescentando programa*

```
15 LET total=0
```

Flores

20 FOR a=1 TO 5

PLOT e DRAW

10 PLOT 100,100: DRAW 100,50



## CONHEÇA OS LIVROS DA EDITORA CAMPUS LTDA.

**SIM • Estou interessado**

Desejo receber informações dos lançamentos da Editora Campus nas seguintes áreas\*:

- |                                                      |                                      |                                           |
|------------------------------------------------------|--------------------------------------|-------------------------------------------|
| <input type="checkbox"/> INFORMÁTICA                 | <input type="checkbox"/> ESTATÍSTICA | <input type="checkbox"/> ECONOMIA         |
| <input type="checkbox"/> MATEMÁTICA                  | <input type="checkbox"/> FÍSICA      | <input type="checkbox"/> ADMINISTRAÇÃO    |
| <input type="checkbox"/> ENG <sup>o</sup> ELETRÔNICA | <input type="checkbox"/> QUÍMICA     | <input type="checkbox"/> CIÊNCIAS SOCIAIS |
| <input type="checkbox"/> ENG <sup>o</sup> MECÂNICA   | <input type="checkbox"/> PSICOLOGIA  | <input type="checkbox"/> ARTE             |
| <input type="checkbox"/> ENG <sup>o</sup> CIVIL      | <input type="checkbox"/> PSICANÁLISE | <input type="checkbox"/> OUTRAS           |

*Favor indicar com dois X, no máximo, até duas áreas de interesse prioritário e com um X as áreas de interesse secundário*

Livro Adquirido.....

Livraria..... Estado.....

Nome.....

Endereço.....

Estado..... Cidade..... CEP.....

Profissão..... Função.....

Instituição.....

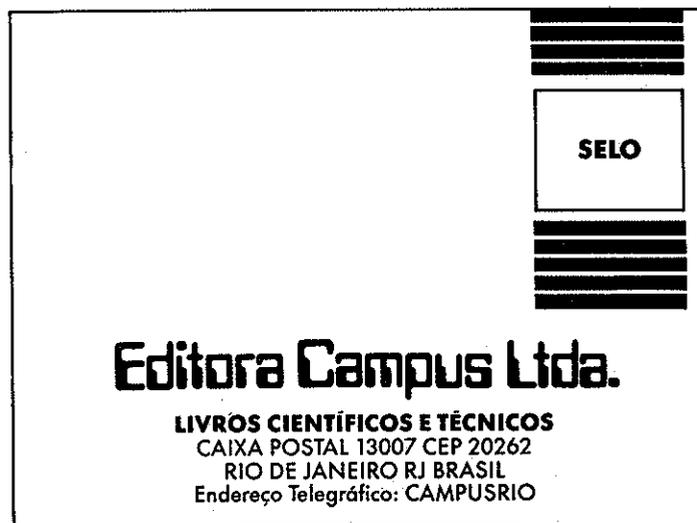
**IMPORTANTE: É imprescindível a indicação de Livro Adquirido e Profissão.**



**NOSSOS LIVROS ENCONTRAM-SE  
EM TODAS AS BOAS LIVRARIAS**

**DOBRE AQUI**

---



**Editora Campus Ltda.**

**LIVROS CIENTÍFICOS E TÉCNICOS**  
CAIXA POSTAL 13007 CEP 20262  
RIO DE JANEIRO RJ BRASIL  
Endereço Telegráfico: CAMPUSRIO

**COLE**